DECOMPOSITION AND INVERSION OF CONVOLUTION OPERATORS

By

ZOHRA Z. MANSEUR

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1990

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Gerhard X. Ritter, first for coming up with such a great concept as the Image Algebra, and second for allowing me to be part of his growing team. His guidance is also greatly appreciated.

This dissertation would not have been possible without the constant help of my other advisor, Dr. David C. Wilson. His inquisitive mind encouraged me to always look further. I would like to thank him for being there, available, patient, and helpful. The generous use of his office space and facilities greatly helped in the completion of this work.

Drs. James E. Keesling, Li-Chien Shen, and Joseph N. Wilson deserve thanks for their service as members of my committee. I would like to thank Dr. Keesling also for his willingness to serve as one of my references.

The financial support I received from the Mathematics Department in the form of a teaching assistanship, and from Dr. Ritter in the from of a research assistanship is gracefully ackowleged. I take this opportunity to thank the secretaries of the Mathematics Department for caring and helping, when needed. I wish to thank also Randy Fischer for all the help on computers.

I am deeply grateful to my parents, for providing me with a first class education, despite the many difficulties they encountered in their life. My mother, Faiza Houasnia, and my father, Tahar Zbiri, have taught me that a woman's independence

can be achieved mainly through her education. Special thanks go to my parents-in-law Dhia Si-Ahmed, and Omar Manseur for their support and trust, and for raising such a fine son.

Very sweet thoughts go to my husband and best friend, Rachid, for his understanding, and his patience throughout our marriage. He believed in me, as much as my parents did. His constant help with computers gave me the privilege to have almost all my technical questions answered. My children, Mehdi and Maya, made my student life harder sometimes, but they also made it all worthwhile.

Finally, I wish to thank all my friends. They made Gainesville a very enjoyable city to live in.

TABLE OF CONTENTS

LIST OF SYMBOLS

$\mathbf{Z}$:	the set of integers
$\mathbf{R}$:	the set of real numbers
$\mathbf{R}^{\mathrm{n}}$:	n-dimensional Euclidean vector space
$\mathbf{F}$:	an arbitrary value set
$\in, \notin, \subset$:	is an element, is not an element, is a subset of
$\mathbf{X}$, $\mathbf{Y}$:	point sets (in particular $m \times n$ arrays)
$\mathbf{X} \setminus \mathbf{Y}$:	the set difference of $\mathbf{X}$ and $\mathbf{Y}$
$\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$:	points
$\mathbf{a} : \mathbf{X} \to \mathbf{F}$:	$\mathbf{a}$ is an $\mathbf{F}$ valued image on $\mathbf{X}$
$\mathbf{a}$, $\mathbf{b}$:	images
$\mathbf{a} + \mathbf{b}$:	addition of image $\mathbf{a}$ and image $\mathbf{b}$
$\mathbf{F}^{\mathbf{X}}$:	the set of $\mathbf{F}$ valued images on $\mathbf{X}$
$\mathbf{t} : \mathbf{Y} \to \mathbf{F}^{\mathbf{X}}$:	$\mathbf{t}$ is an $\mathbf{F}$ valued template from $\mathbf{Y}$ to $\mathbf{X}$
$\mathbf{t}_{\mathbf{y}}$:	the template $\mathbf{t}$ at the point $\mathbf{y}$
$S(\mathbf{t}_{\mathbf{y}})$:	the support of the template $\mathbf{t}$ at the point $\mathbf{y}$
$\mathbf{r}$, $\mathbf{s}$, $\mathbf{t}$:	templates
$(\mathbf{F}^{\mathbf{X}})^{\mathbf{Y}}$:	the set of $\mathbf{F}$ valued templates from $\mathbf{Y}$ to $\mathbf{X}$
$\mathbf{L}_{\mathbf{X}}$:	the set $(\mathbf{F}^{\mathbf{X}})^{\mathbf{X}}$
$\mathbf{F}[y_1, y_2, ..., y_n]$:	the ring of multivariate polynomials over $\mathbf{F}$

$\mathbf{R}[x,y]/(x^m - 1, y^n - 1)$:	the quotient ring of polynomials modulo $x^m - 1$ and $y^n - 1$
$M_n(\mathbf{F})$:	the algebra of $n \times n$ matrices over $\mathbf{F}$
$\psi, \ \eta, \ \theta$:	isomorphisms
ω_n:	$\exp(-2\pi i/n)$, where $i = \sqrt{(-1)}$
$\oplus$:	generalized convolution
$\mathrm{cir}(c_0, c_1, ..., c_n)$:	an $n \times n$ circulant matrix
$\mathbf{C_{mn}}$:	the set of circulant matrices
$\mathbf{F_n}$:	the one-dimensional Fourier matrix of order n
$\mathbf{F_{mn}}$:	the two-dimensional Fourier matrix of order $m \times n$

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

DECOMPOSITION AND INVERSION OF CONVOLUTION OPERATORS

By

Zohra Z. Manseur

December 1990

Chairman: Dr. Gerhard X. Ritter
Co-Chairman: Dr. David C. Wilson
Major Department: Mathematics

The image algebra, an algebraic structure developed by G. X. Ritter et. al. specifically to meet the needs of digital image processing, will provide the mathematical setting for this investigation. The results presented in this dissertation evolved as a direct result of questions that arose during the development of this image algebra. A network of processors for a parallel computer architecture is modeled as a subset of $\mathbf{R}^{\mathbf{n}}$. An image is represented as a function defined on such a subset. The

subsets of concern, here, will be rectangular arrays with m rows and n columns. A template will be represented by a function from points in a finite rectangular array into images. Basic operands and operations of the image algebra, such as addition, multiplication and convolution, are defined. Relationships between image algebra, matrix (linear) algebra, and polynomial algebra are described. The two main questions addressed in this dissertation are the problems of template decomposition and template inversion.

Methods are established for a two-dimensional shift invariant convolution operator defined on a rectangular array to be decomposed into sums and products of operators of smaller size. The main result of this type is that 5×5 shift invariant operators can be written as the sum and product of at most five 3×3 shift invariant operators. The second is that a 5×5 operator satisfying certain symmetry conditions can be written as the sum and product of at most three 3×3 operators. The decomposition methods presented here are suitable for a machine with a pipeline architecture.

An additional focus of this research will center on the problem of extending results valid for shift invariant operators to non-shift invariant operators. In particular, necessary and sufficient conditions are given for a variant template to be separable. A 5×5 variant template can always be written as the sum and product of at most seven 3×3 templates.

The mean filter with five nonzero weights all equal to one and whose configuration (or shape) is in the form of a cross is shown to be invertible when defined on an $m \times m$ array if and only if neither 5 nor 6 divide m.

CHAPTER 1

INTRODUCTION

This dissertation is the result of work done in conjunction with an investigation of the structure of the image algebra, an algebraic structure for use in image processing. The need for a unifying mathematical theory for image processing algorithms specification, and a more powerful basis for an algebraically-based, high level programming language, led to the development of the image algebra. It is an algebra that operates on images, and whose operators reflect the types of transformations commonly used in digital image processing [2]. It is capable of expressing all types of transformations and algorithms commonly used in image processing, in terms of its operators [28,38].

G. Matheron [18] and J. Serra [32] initiated the use of image algebra in image processing. It was known then as mathematical morphology. Sternberg [33,34] also investigated it independently. This algebra was based on the operations of Minkowski addition and subtraction of sets in $\mathbf{R}^n$ [24]. Mathematical morphological is about the study of shapes and patterns. The operations of Minkowski are often referred to as dilation and erosion operations. Theses morphological operations techniques can be applied to many image processing and image analysis problems. However, they cannot, with the exceptions of very few simple cases, be used as a basis for a general purpose algebraically based language for digital image processing. Their weakness resides in the fact that they cannot easily express many commonly

used transformations such as linear image to image transformations and algorithms [19]. In response to these limitations, and in order to establish an algebraic structure capable of expressing most of the common image processing algorithms, G. X. Ritter at the University of Florida began the development of a more general image algebra capable of expressing all linear and morphological transformations [23,25,26,28].

This image algebra, will provide the mathematical setting for most of the results presented in this dissertation. In Chapter 2 basic operands and operations are defined, and relationships between image algebra linear algebra and polynomial algebra are discussed. Isomorphisms imbedding the linear algebra, and the polynomial algebra into the image algebra give rise to the observation that a linear (or generalized) convolution is equivalent to matrix multiplication, and the observation that operator inversion (or deconvolution) is equivalent to matrix inversion. In the case that an operator is circulant, operator inversion is equivalent to polynomial inversion. In the usual implementation of the isomorphism between templates defined on rectangular arrays and matrices, a shift invariant operator corresponds to a block Toeplitz matrix with Toeplitz blocks, and a circulant operator corresponds to a block circulant matrix with circulant blocks. The image algebra allows us to give a precise formulation of the decomposition methods for non-shift invariant operators. Recall that some frequently used non-shift-invariant operators include the Fourier Transform or the Gabor Transform [7,11].

The use of parallel processing has come to play an increasingly important role in image processing during the past few years. Efficient implementation of linear convolution is an important aspect of any hardware environment. Some classical examples of two-dimensional operators include the Gaussian mean filter, the Laplacian filter, and the Discrete Fourier Transform.

Template decomposition plays a fundamental role in image processing algorithm optimization, since it provides means for reducing the cost in computation time, and therefore increases the comptutational efficiency of image processing algorithms. This goal can be achieved in two ways, either by reducing the number of arithmetic computations in an algorithm or by restructuring the algorithms to match the structure of a special image processing architecture. One of the goals of this work is to present methods for the decomposition of a two-dimensional shift-invariant convolution operator in a form suitable for a parallel image processing machine with hardware similar to the PIPE developed at the Center for Manufacturing Engineering of the National Bureau of Standards. As mentioned by O'Leary [22], a machine of this type is able to efficiently apply a 3×3 convolution operator to a 256×256 image, but has a more difficult time applying a 5×5 operator. Another example of a *pipeline* computer is ERIM's CytoComputer [34] which can deal only with templates of size 3×3 or smaller on each pipeline stage. Thus, unless Fourier Transform methods are to be used, operators of size larger than 3×3 must be decomposed into sums and products of 3×3 operators to fit the hardware restriction. O'Leary showed that if an operator (viewed as a matrix) has rank n, then it can be written as the sum of at most n rank 1 (i.e. separable) operators. An immediate consequence of the rank one method described by O'Leary in [22] is that any arbitrary 5×5 shift-invariant operator can be written in the form

$$\mathbf{t} = (\mathbf{t}_1 \oplus \mathbf{t}_2) + (\mathbf{t}_3 \oplus \mathbf{t}_4) + (\mathbf{t}_5 \oplus \mathbf{t}_6) + (\mathbf{t}_7 \oplus \mathbf{t}_8) + (\mathbf{t}_9 \oplus \mathbf{t}_{10}),$$

where each $\mathbf{t}_i$ is a 3×3 shift-invariant operator and $\oplus$ denotes the convolution operation.

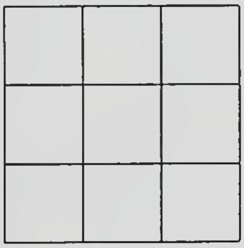
While the results on shift invariant operators are presented in Chapter 3 as theorems concerning the decomposition of polynomials of two variables into sums and products of lower degree polynomials, the theorems discussed in Chapter 4 will

all be phrased in terms of sums and $\oplus$ operations on templates. In Chapter 3, it is proved that an arbitrary 5×5 shift-invariant template $\mathbf{t}$ defined on $\mathbf{Z} \times \mathbf{Z}$ can be decomposed into the form

$$\mathbf{t} = (\mathbf{t}_1 \oplus \mathbf{t}_2) + (\mathbf{t}_3 \oplus \mathbf{t}_4) + \mathbf{t}_5,$$

where each $\mathbf{t}_i$ is a 3×3 shift-invariant template. Thus, the number of 3×3 templates needed to decompose $\mathbf{t}$ is actually half of that indicated by the rank method. Example 5 of Chapter 5 shows that this decomposition is the best possible, in the sense that it may not always be possible to decompose an operator into the sum and product of fewer than five 3×3 templates.

The most commonly used template in the decompositions is the 3×3 square template, better known as the *Moore* neighborhood. It has the following form:



Since most templates used in image processing are symmetric or skew symmetric with respect to the x or the y axis, a discussion of the problem of decomposing convolution operators exhibiting some type of symmetry is included in Chapter 3 and Chapter 4.

An second approach to the decomposition of two dimensional operators based on the lower-upper triangular (LU) factorization of a rectangular matrix, will result into the finite sum of separable operators. A comparison of the polynomial method and the LU factorization methods is given in Chapter 3 Section 4.

Results concerning the decomposition of arbitrary operators, (i.e. those which may not be shift invariant) are presented in Chapter 4. Necessary and sufficient conditions for a variant operator to be separable, i.e. splittable into the convolution of a horizontal operator and a vertical operator, are established. Since not all templates are separable, we will show that every variant template can be written as the sum of separable templates.

We conclude by pointing out that template decomposition is not only necessary in the case where a special image processing device cannot handle large templates but is also desirable when the problem of efficient decomposition is concerned in real-time applications.

Inverse problems have come to play a central role in modern applied mathematics. While classical linear and non-linear inverse problems abound in mathematical physics [31], they are also important in such imaging areas as tomography [31], remote sensing [31], and restoration [30]. While Rosenfeld and Kak [30] showed how the Wiener filter and least square methods can be used to restore an image, they also explained the equivalence between these techniques and the problem of inverting a block circulant matrix with circulant blocks. In 1973, G. E. Trapp [36] showed how the Discrete Fourier Transform could be used to diagonalize and invert a matrix that was either circulant or block circulant with circulant blocks. While the algebraic relationship between circulant matrices and polynomials was completely formulated by J. P. Davis [6], it was P. D. Gader and G. X .Ritter [9,27] who made the connection between polynomials and circulant templates. In particular, it was found that a circulant template defined on a rectangular array with m rows and n columns is invertible if and only if its corresponding polynomial $p(x, y)$ has the property that $p(\omega_n^j, \omega_m^k) \neq 0$, for all $0 \leq j \leq n$, and $0 \leq k \leq m$. (The symbol ω_n denotes the root of the unity, $\exp(2\pi i/n)$.)

An application of this method is that the usual 3×3 mean filter defined on a rectangular array with m rows and n columns is invertible if and only if the number 3 does not divide either m or n. Thus for example, the mean filter defined on an array with 512 rows and 512 columns will be invertible, while the mean filter defined on an array with 512 rows and 240 columns will not be invertible. Gader [8] asked if a similar result can be obtained for the *von Neumann* mean filter. The *von Neumann* mean filter is the template defined by

$$\mathtt{t} = \begin{array}{|c|c|c|} \cline{2-2} \multicolumn{1}{c|}{} & 1 & \multicolumn{1}{c}{} \\ \hline 1 & \langle 1 \rangle & 1 \\ \hline \multicolumn{1}{c|}{} & 1 & \multicolumn{1}{c}{} \\ \cline{2-2} \end{array} .$$

The main result of Chapter 6 is to prove that if $\mathbf{t}$ is defined on a square array with m rows and m columns, then $\mathbf{t}$ is invertible if and only if neither 5 nor 6 divide m.

A variety of examples are presented in chapter 4 to both illustrate the utility of the methods developed and to demonstrate the fact that some theorems are best possible.

CHAPTER 2

IMAGE ALGEBRA: AN OVERVIEW

2.1. Introduction

The following chapter contains a brief review of the fundamental concepts and notation of the image algebra. We will need these ideas to extend results valid for shift-invariant operators to more general operators. For the purpose of this dissertation, only basic definitions and operations are given. The relationship between image algebra and linear algebra will also be described in this chapter. For a full and detailed description of all image algebra operands and operations, see [29].

The image algebra is a heterogeneous algebraic structure specially designed to meet the needs of digital image processing. It is used as a model and tool for the development of local, parallel algorithms. Several commonly used image processing transformations, such as general convolutions, Discrete Fourier Transform, and edge detection techniques can be expressed in terms of the image algebra.

The use of image algebra in digital image processing was initiated by Serra [32], Miller [19], and Sternberg [33]. It was Ritter [24] who showed that their algebras were all equivalent and based on the morphological operations of Minkowski.

An image algebra is an algebra whose operands are images and subimages (or neighborhoods). It deals with six basic type of operands, namely, value sets,

coordinate sets (or point sets), the elements of the value and the coordinate sets, images and templates.

2.2. Definitions and Background

2.2.1. Operands of the Image Algebra

A *value set* is the set of values that an image can take. It can be any semi-group with a zero element. The most commonly used ones in image processing are the set of positive integers $\mathbf{Z}^+$, integers $\mathbf{Z}$, rational numbers $\mathbf{Q}$, real numbers $\mathbf{R}$, positive real numbers $\mathbf{R}^+$, or complex numbers $\mathbf{C}$. An unspecified value set will be denoted by $\mathbf{F}$.

A *coordinate set (or a point set)* is a subset of an n-dimensional Euclidean space, $\mathbf{R}^{\mathbf{n}}$, for some n. It is commonly denoted by $\mathbf{X}$ or $\mathbf{Y}$, and the elements of such sets are denoted by lower case letters. The familiar point sets are the rectangular and hexagonal arrays.

DEFINITION 2.1. *Let* $\mathbf{X}$, *and* $\mathbf{F}$ *be a point and a value set, respectively. An* $\mathbf{F}$ *valued image* $\mathbf{a}$ *on* $\mathbf{X}$ *is a function* $\mathbf{a} : \mathbf{X} \to \mathbf{F}$.

Thus, the graph of an $\mathbf{F}$ valued image $\mathbf{a}$ on $\mathbf{X}$ is of the form

$$\mathbf{a} = \{(\mathrm{x}, \mathbf{a}(\mathrm{x})) : \mathbf{a}(\mathrm{x}) \in \mathbf{F}, \text{ for all } \mathrm{x} \in \mathbf{X}\}.$$

The set $\mathbf{X}$ is called the *set of image points* of $\mathbf{a}$, and the range of the function $\mathbf{a}$ is called the *set of image values* of $\mathbf{a}$. The pair $(\mathrm{x}, \mathbf{a}(\mathrm{x}))$ is called a *picture element* or a *pixel* , x the *pixel location* , and $\mathbf{a}(\mathrm{x})$ the *pixel (or gray) value* . We will denote

the set of all $\mathbf{F}$ valued images on $\mathbf{X}$ by $\mathbf{F}^{\mathbf{X}}$. We make no distinction between an image and its graph.

DEFINITION 2.2. *An image* $\mathsf{a} : \mathbf{X} \to \mathbf{F}$ *has finite support on* $\mathbf{X}$ *if* $\mathsf{a}(\mathrm{x}) \neq 0$ *for only a finite number of elements* $\mathrm{x} \in \mathbf{X}$.

Another basic, but very powerful tool of the image algebra, is the generalized template.

DEFINITION 2.3. *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be two coordinate sets, and let* $\mathbf{F}$ *be a value set. A generalized* $\mathbf{F}$ *valued template* t *from* $\mathbf{Y}$ *to* $\mathbf{X}$ *is a function* $\mathsf{t} : \mathbf{Y} \to \mathbf{F}^{\mathbf{X}}$.

Thus, for each $\mathrm{y} \in \mathbf{Y}, \mathsf{t}(\mathrm{y}) \in \mathbf{F}^{\mathbf{X}}$, or equivalently, $\mathsf{t}(\mathrm{y})$ is an $\mathbf{F}$ valued image on $\mathbf{X}$. For notational convenience, we define $\mathsf{t}_{\mathrm{y}} \equiv \mathsf{t}(\mathrm{y})$. Thus,

$$\mathsf{t}_{\mathrm{y}} = \{(\mathrm{x}, \mathsf{t}_{\mathrm{y}}(\mathrm{x})) : \mathrm{x} \in \mathbf{X}\}.$$

The sets $\mathbf{Y}$ and $\mathbf{X}$ are called the *target domain* and *range space* of t, respectively. The point y is called the *target (or domain) point* of the template t, and the values $\mathsf{t}_{\mathrm{y}}(\mathrm{x})$ are called the *weights* of the template t at y. Note that the set of all $\mathbf{F}$ valued templates from $\mathbf{Y}$ to $\mathbf{X}$ can be denoted by $(\mathbf{F}^{\mathbf{X}})^{\mathbf{Y}}$.

If t is a template from $\mathbf{Y}$ to $\mathbf{X}$, then the set

$$\mathcal{S}(\mathsf{t}_{\mathrm{y}}) = \{\mathrm{x} \in \mathbf{X} : \mathsf{t}_{\mathrm{y}}(\mathrm{x}) \neq 0\}$$

is called the *support* of t_{y}.

If t is an $\mathbf{F}$ valued template from $\mathbf{X}$ to $\mathbf{X}$, and $\mathbf{X}$ is a subset of $\mathbf{R}^{\mathrm{n}}$, then t is called *translation invariant (or shift-invariant)* if and only if for each triple $\mathrm{x}, \mathrm{y}, \mathrm{z} \in \mathbf{X}$, with $\mathrm{x} + \mathrm{z}$ and $\mathrm{y} + \mathrm{z} \in \mathbf{X}$, we have that

$$t_y(x) = t_{y+z}(x + z).$$

Note that a translation invariant template must be an element of $(\mathbf{F}^{\mathbf{X}})^{\mathbf{X}}$. Invariant operators on $\mathbf{Z} \times \mathbf{Z}$ are commonly expressed in terms of polynomials of two variables. For brevity, we will frequently refer to shift-invariant templates as invariant templates. A template which is not necessarily translation invariant is called translation variant or, simply, a variant template. Translation invariant templates occur naturally in digital image processing.

2.2.2. Operations of the Image Algebra

The basic operations on and between $\mathbf{F}$ valued images are naturally derived from the algebraic structure of the value set $\mathbf{F}$.

Let $\mathbf{X}$ be a subset of $\mathbf{R}^n$. Suppose $\mathbf{a} \in \mathbf{R}^{\mathbf{X}}$ and $\mathbf{t} \in (\mathbf{R}^{\mathbf{X}})^{\mathbf{X}}$.

Addition on images is defined as follows: If $\mathbf{a}, \mathbf{b} \in \mathbf{F}^{\mathbf{X}}$, then

$$\mathbf{a} + \mathbf{b} \equiv \{(x, \mathbf{c}(x)) : \mathbf{c}(x) = \mathbf{a}(x) + \mathbf{b}(x), \quad x \in \mathbf{X}\}.$$

Higher level operations are the ones that involve operations between templates and images, and between templates only.

The addition of two templates is defined pointwise. If $\mathbf{s}$ and $\mathbf{t} \in (\mathbf{R}^{\mathbf{X}})^{\mathbf{X}}$, then we have

$$(\mathbf{s} + \mathbf{t})_y(x) = \mathbf{s}_y(x) + \mathbf{t}_y(x).$$

DEFINITION 2.4. *The generalized convolution of an image* $\mathbf{a}$ *with a template* $\mathbf{t}$ *is defined by*

$$\mathbf{a} \oplus \mathbf{t} \equiv \{(y, \mathbf{b}(y)) : \mathbf{b}(y) = \sum_{x \in \mathbf{X}} \mathbf{a}(x)\mathbf{t}_y(x), \quad y \in \mathbf{X}\}.$$

Linear convolution plays a fundamental role in image processing. It is involved in as many important examples as the Discrete Fourier Transform, the Laplacian, the mean or average filter and the Gaussian mean filter.

DEFINITION 2.5. *If* $\mathbf{s}$ *and* $\mathbf{t}$ *are templates on* $\mathbf{X}$, *then we define the generalized convolution of the two templates as the template* $\mathbf{r} = \mathbf{s} \oplus \mathbf{t}$ *by defining each image function* $\mathbf{r_y}$ *by the rule*

$$\mathbf{r_y} = \{(\mathbf{z}, \mathbf{r_y}(\mathbf{z})) : \mathbf{r_y}(\mathbf{z}) = \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{t_y}(\mathbf{x}) \mathbf{s_x}(\mathbf{z}), \ \textit{where } \mathbf{z} \in \mathbf{X}\}.$$

Note that $\mathbf{r}$ can be viewed as a generalization of the usual notion of the composition of two convolution operators. If the templates $\mathbf{r}$ and $\mathbf{s}$ are translation invariant, then except for values near the boundary, the previous definition agrees with the usual definition of polynomial product.

Note also that if $\mathbf{s}$ and $\mathbf{t}$ are two invariant templates, then $\mathbf{r}$ would be an invariant template too. Defining $\mathbf{r}$ at an arbitrary $\mathbf{y} \in \mathbf{Y}$ is sufficient to define the template everywhere.

Many other image operations are described in detail in Ritter et al [29]. A precise investigation of the linear operator $\oplus$ can also be found in Gader [8], and an extensive study of other non-linear template operations can be found in Davidson [5] and Li [16].

2.3. Image Algebra and Linear Algebra

If $\mathbf{X}$ is a finite rectangular subset of the plane with m rows and n columns, then it can be linearly ordered left to right and row by row. Thus, we can write $\mathbf{X} =$

$\{x_1, x_2, ..., x_{mn}\}$. Let $L_{\mathbf{X}}$ denote the set of templates on $\mathbf{X}$. (i.e. $L_{\mathbf{X}} = (\mathbf{F}^{\mathbf{X}})^{\mathbf{X}}$.)

Let $(M_{mn}, +, *)$ denote the ring of $mn \times mn$ matrices with entries from $\mathbf{F}$ under matrix addition and multiplication. For any template $\mathbf{t}$, we define a matrix $M_{\mathbf{t}} = (m_{ij})$ where $m_{ij} = \mathbf{t}_{\mathbf{x}_j}(\mathbf{x}_i)$. For the sake of notational convenience we will write t_{ji} for $\mathbf{t}_{\mathbf{x}_j}(\mathbf{x}_i)$.

Define the mapping $\psi : L_{\mathbf{X}} \to M_{mn}$ by $\psi(\mathbf{t}) = M_{\mathbf{t}}$.

The next Theorem was proved by Ritter and Gader [8]. It shows that there is an embedding of linear algebra into image algebra.

THEOREM 2.6. *The mapping ψ is an isomorphism from the ring $(L_{\mathbf{X}}, +, \oplus)$ onto the ring $(M_{mn}, +, *)$. That is, if $\mathbf{s}, \mathbf{t} \in L_{\mathbf{X}}$, then*

1.) $\psi(\mathbf{s} + \mathbf{t}) = \psi(\mathbf{s}) + \psi(\mathbf{t})$ *or* $M_{\mathbf{s}+\mathbf{t}} = M_{\mathbf{s}} + M_{\mathbf{t}}$,

2.) $\psi(\mathbf{s} \oplus \mathbf{t}) = \psi(\mathbf{s})\psi(\mathbf{t})$ *or* $M_{\mathbf{s}\oplus\mathbf{t}} = M_{\mathbf{s}}M_{\mathbf{t}}$,

3.) ψ *is one-to-one and onto.*

This theorem clearly states that template inversion or deconvolution is equivalent to matrix inversion. Actually, a more powerful implication of this theorem is that any tool available in linear algebra can be directly applicable to problems in image algebra.

DEFINITION 2.7. *Let $\mathbf{X}$ be an $m \times n$ coordinate set. We say that the mapping $\phi : \mathbf{X} \to \mathbf{X}$ is a circulant translation if and only if ϕ is of the form $\phi(\mathbf{x} + \mathbf{h}) = (\mathbf{x} + \mathbf{h})(mod(m, n))$, for some $\mathbf{h} \in \mathbf{X}$.*

DEFINITION 2.8. *We say that* $\mathbf{t} \in (\mathbf{F}^{\mathbf{X}})^{\mathbf{X}}$ *is circulant if and only if for every circulant translation* ϕ, *the equation* $\mathbf{t}_{\mathbf{x}}(\mathbf{y}) = \mathbf{t}_{\phi(\mathbf{x})}(\phi(\mathbf{y}))$ *holds.*

This last definition shows that a circulant template is completely determined if it is defined at only one point. Translation invariant and circulant templates are used in the implementation of the convolution and therefore, are directly related to the Discrete Fourier Transform.

For the purposes of this dissertation, $\mathbf{X}$ will usually be a subset of $\mathbf{Z} \times \mathbf{Z}$, where $\mathbf{Z}$ denotes the set of integers. All the templates we will consider will have finite support contained in a set of the form

$$\mathbf{X}_{m,n} = \{(i,j) : |i| \leq m \text{ and } |j| \leq n\}.$$

Therefore, a $(2m+1) \times (2n+1)$ template on $\mathbf{Z} \times \mathbf{Z}$ will have support contained in $\mathbf{X}_{\mathbf{m},\mathbf{n}}$ and will be displayed as a matrix of the form

$$\mathbf{t} = \begin{bmatrix} t_{-m,-n} & \cdots & t_{-m,0} & \cdots & t_{-m,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ t_{0,-n} & \cdots & < t_{0,0} > & \cdots & t_{0,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ t_{m.-n} & \cdots & t_{m,0} & \cdots & t_{m,n} \end{bmatrix}.$$

If the template $\mathbf{t}$ is defined as above, then it is understood that all weights $t_{ij} = 0$, for all (i,j) such that $|i| > m$ or $|j| > n$. The element $< t_{0,0} >$ will denote the weight at the center pixel location (i.e. $\mathbf{t}_x(x) = t_{0,0}$). Note that the indexing here differs by a shift from the one given above.

CHAPTER 3

SHIFT INVARIANT OPERATORS

The use of parallel processing has been increasing for the past years. Linear convolution is widely used in image processing. It consists of applying an operation between a template and a given input image, pixelwise, to yield an output image.

One of the main reasons for template decomposition is that some current image processors can only directly implement operations on very small templates . Most transforms are not able to be implemented directly on a parallel architecture. Consider, for example, a parallel image processing machine with hardware similar to the PIPE, developed at the Center for Manufacturing Engineering of the National Bureau of Standards. This machine is capable of evaluating a small template , like a 3×3,but cannot easily evaluate a template of larger size.

The problem of template decomposition is to try to find a sequence of templates $\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_n$ smaller in size than the given template $\mathbf{t}$, such that when we apply an operation on template $\mathbf{t}$ and image $\mathbf{a}$, it will be equivalent to sequentially applying operations between $\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_n$ to the image. For example, in the case of the neighborhood array processor, it will consist of decomposing the template into a sequence of factors, where each factor is directly implementable on the architecture.

The motivation behind all the work on template decomposition is to increase the speed and reduce the convolution computation time. If a convolution operator is large in size, then the cost in computation time can be prohibitive. This compu-

tational complexity can be reduced if the given template in decomposed into sums and convolutions of smaller size templates.

The following discussion will clarify the use of template decomposition. Suppose a template **t** has the following decomposition:

$$\mathbf{t} = (\oplus_{i=1}^{i=h} \mathbf{r}_i) + (\oplus_{j=1}^{j=k} \mathbf{s}_j).$$

The convolution of an image **a** together with the template **t** will give

$$\mathbf{a} \oplus \mathbf{t} = \mathbf{a} \quad \oplus \quad (\oplus_{i=1}^{i=h} \mathbf{r}_i + \oplus_{j=1}^{j=k} \mathbf{s}_j)$$

$$= [(...((\mathbf{a} \oplus \mathbf{r}_1) \oplus \mathbf{r}_2)...) \oplus \mathbf{r}_h] \quad + \quad [(...((\mathbf{a} \oplus \mathbf{s}_1) \oplus \mathbf{s}_2)...) \oplus \mathbf{s}_k]$$

$$= \mathbf{b} \quad \oplus \quad \mathbf{c},$$

where $\mathbf{b} = [(...((\mathbf{a} \oplus \mathbf{r}_1) \oplus \mathbf{r}_2)...) \oplus \mathbf{r}_{\mathrm{h}}]$, and $\mathbf{c} = [(...((\mathbf{a} \oplus \mathbf{s}_1) \oplus \mathbf{s}_2)...) \oplus \mathbf{s}_{\mathrm{k}}]$.

When $\mathbf{a} \oplus \mathbf{t}$ is computed, the image **b** is computed first, the result stored, then the image **c** is computed and the sum of the two is taken.

3.1. Decomposition Methods for General Operators

Templates come in different weights, size or shape, depending on the image processing application. A special classs of invariant templates are the two dimensional *rectangular* templates. The support of a rectangular template is a rectangular array.

While a 5×5 operator is usually defined as a matrix of the form

$$\begin{bmatrix} t_{-2,2} & t_{-1,2} & t_{02} & t_{12} & t_{22} \\ t_{-2,1} & t_{-1,1} & t_{01} & t_{11} & t_{21} \\ t_{-2,0} & t_{-1,0} & < t_{00} > & t_{10} & t_{20} \\ t_{-2,-1} & t_{-1,-1} & t_{0,-1} & t_{1,-1} & t_{2,-1} \\ t_{-2,-2} & t_{-1,-2} & t_{0,-2} & t_{1,-2} & t_{2,-2} \end{bmatrix},$$

we will express all the theorems, corollaries, and propositions in terms of polynomials. To avoid negative exponents, we begin all indexing with the integer zero. This implies, if $n = m = 4$, that the remainder term $r(x, y)$ is shifted, so that its value at the center pixel location will be represented by r_{22}. Thus, in the decomposition methods presented in Sections 2, 3, and 4, it will never be necessary to shift the image before applying the operator $r(x, y)$.

THEOREM 3.1. *If $t(x, y)$ is a polynomial defined by the formula*

$$t(x, y) = \sum_{i=0}^{m} \sum_{j=0}^{n} t_{ij} x^i y^j,$$

then there exist five polynomials $p_1(x, y)$, $p_2(x, y)$, $q_1(x, y)$, $q_2(x, y)$ and $r(x, y)$ such that

$$t(x, y) = p_1(x, y) q_1(x, y) + p_2(x, y) q_2(x, y) + r(x, y),$$

where

$$p_1(x, y) = \sum_{i=0}^{2} \sum_{j=0}^{2} p_{1ij} x^i y^j, \quad q_1(x, y) = \sum_{i=0}^{m-2} \sum_{j=0}^{n-2} q_{1ij} x^i y^j,$$

$$p_2(x, y) = \sum_{i=0}^{2} \sum_{j=0}^{2} p_{2ij} x^i y^j, \quad q_2(x, y) = \sum_{i=0}^{m-2} \sum_{j=0}^{n-2} q_{2ij} x^i y^j,$$

and

$$r(x,y) = \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} r_{ij}x^i y^j.$$

PROOF:

If $a(y) = \sum_{i=0}^{n} t_{0i}y^i$, then by the Fundamental Theorem of Algebra, there are two polynomials $a_1(y)$, and $a_2(y)$ with $\deg(a_1(y)) \leq 2$ and $\deg(a_2(y)) \leq n-2$, such that $a(y) = a_1(y)a_2(y)$. Similarly, if $c(y) = \sum_{i=0}^{n} t_{mi}y^i$, then there are two polynomials $c_1(y)$, and $c_2(y)$ with $\deg(c_1(y)) \leq 2$ and $\deg(c_2(y)) \leq n-2$, such that $c(y) = c_1(y)c_2(y)$.

If we choose

$$p_1(x,y) = a_1(y) + x^2 c_1(y),$$
$$q_1(x,y) = a_2(y) + x^{m-2}c_2(y),$$

then the polynomial

$$\begin{aligned} u(x,y) &= t(x,y) - p_1(x,y)q_1(x,y) \\ &= \sum_{i=0}^{m}\sum_{j=0}^{n} u_{ij}x^i y^j \end{aligned}$$

will have the property that if u_{ij} is nonzero, then $1 \leq i \leq m-1$.

If $b(x) = \sum_{i=1}^{m-1} u_{i0}x^i$, then the polynomial $b(x)$ can be decomposed as $xb_1(x)$, where $\deg(b_1(x) \leq m-2$. Similarly, if $d(x) = \sum_{i=1}^{m-1} u_{in}x^i$, then $d(x)$ can be decomposed as $xd_1(x)$, where $\deg(d_1(x)) \leq m-2$.

Let

$$p_2(x,y) = x + xy^2, \text{ and}$$
$$q_2(x,y) = b_1(x) + d_1(x)y^{n-2}.$$

If

$$\begin{aligned} r(x,y) &= u(x,y) - p_2(x,y)q_2(x,y) \\ &= \sum_{i=0}^{m}\sum_{i=0}^{n} r_{ij}x^i y^j, \end{aligned}$$

then when r_{ij} is nonzero, i and j will be such that $1 \leq i \leq m-1$, and $1 \leq j \leq n-3$.

Q.E.D.

COROLLARY 3.2. *If $t(x,y)$ is a polynomial defined by the formula*

$$t(x,y) = \sum_{i=0}^{4}\sum_{j=0}^{4} t_{ij}x^i y^j,$$

then there exist five polynomials $p_1(x,y)$, $p_2(x,y)$, $q_1(x,y)$, $q_2(x,y)$, and $r(x,y)$ such that

$$t(x,y) = p_1(x,y)q_1(x,y) + p_2(x,y)q_2(x,y) + r(x,y),$$

where

$$p_1(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2} p_{1ij}x^i y^j, \quad q_1(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2} q_{1ij}x^i y^j,$$

$$p_2(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2} p_{2ij}x^i y^j, \quad q_2(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2} q_{2ij}x^i y^j,$$

and

$$r(x,y) = \sum_{i=1}^{3}\sum_{j=1}^{3} r_{ij}x^i y^j.$$

Remark 3.3. An immediate consequence of Theorem 3.1 is that if $t(x,y)$ is a polynomial defined by $t(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{n} t_{ij}x^i y^j$, then there exist $n-1$ polynomials $t_1(x,y)$, $t_2(x,y)$,..., $t_{n-2}(x,y)$, and $r(x,y)$ such that

$$t(x,y) = t_1(x,y) + ... + t_{n-2}(x,y) + r(x,y),$$

where each $t_i(x,y)$ is a polynomial which can be written as a product of polynomials of the form

$$p(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2} p_{ij}x^i y^j,$$

and

$$r(x,y) = \sum_{i=1}^{3}\sum_{j=1}^{3} r_{ij}x^i y^j.$$

3.2. A Characterization of Operators Decomposable into a Special Form

THEOREM 3.4. *Let $t(x,y)$ be the polynomial defined by :*

$$t(x,y) = \sum_{i=0}^{m}\sum_{j=0}^{n} t_{ij}x^i y^j,$$

where t_{00}, t_{0n}, t_{m0} and t_{mn} are all nonzero.

Let

$$a(y) = \sum_{i=0}^{n} t_{0i}y^i, \quad b(x) = \sum_{i=0}^{m} t_{in}x^i,$$

$$c(y) = \sum_{i=0}^{n} t_{mi}y^i, \quad \text{and} \quad d(x) = \sum_{i=0}^{m} t_{i0}x^i.$$

Let $\alpha_1, \alpha_2, ..., \alpha_n$ be the roots of $a(y)$; $\beta_1, \beta_2, ..., \beta_m$ be the roots of $b(x)$; $\gamma_1, \gamma_2, ..., \gamma_n$ be the roots of $c(y)$; and $\delta_1, \delta_2, ..., \delta_m$ be the roots of $d(x)$.

If there exists an ordering of the roots so that we have both equations

$$\alpha_1\alpha_2\delta_1\delta_2 = \beta_1\beta_2\gamma_1\gamma_2, \text{ and}$$

$$\alpha_3...\alpha_n\delta_3...\delta_m = \beta_3...\beta_m\gamma_3...\gamma_n$$

(or both equations

$$\alpha_1\alpha_2\beta_1\beta_2 = \gamma_1\gamma_2\delta_1\delta_2, \text{ and}$$

$$\alpha_3...\alpha_n\beta_3...\beta_m = \gamma_3...\gamma_n\delta_3...\delta_m),$$

satisfied, then there exist three polynomials $p(x,y)$, $q(x,y)$ and $r(x,y)$ such that

$$t(x,y) = p(x,y)q(x,y) + r(x,y),$$

where

$$p(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2} p_{ij}x^i y^j, \quad q(x,y) = \sum_{i=0}^{m-2}\sum_{j=0}^{n-2} q_{ij}x^i y^j,$$

and

$$r(x,y) = \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} r_{ij}x^i y^j.$$

Conversely, if

$$t(x,y) = p(x,y)q(x,y) + r(x,y),$$

where $p(x,y), q(x,y)$ *and* $r(x,y)$ *are defined as above, then there exists an ordering of the roots so that we have both equations*

$$\alpha_1\alpha_2\delta_1\delta_2 = \beta_1\beta_2\gamma_1\gamma_2, \text{ and}$$

$$\alpha_3 \ldots \alpha_n \delta_3 \ldots \delta_m = \beta_3 \ldots \beta_m \gamma_3 \ldots \gamma_n$$

(or both equations

$$\alpha_1\alpha_2\beta_1\beta_2 = \gamma_1\gamma_2\delta_1\delta_2, \text{ and}$$

$$\alpha_3 \ldots \alpha_n \beta_3 \ldots \beta_m = \gamma_3 \ldots \gamma_n \delta_3 \ldots \delta_m)$$

satisfied.

PROOF:

Factor $a(y), b(x), c(y)$ and $d(x)$ to obtain

$$\begin{aligned}
a(y) &= t_{0n}(y^n + ... + (t_{01}/t_{0n})y + (t_{00}/t_{0n})) \\
&= t_{0n}(y^2 - (\alpha_1 + \alpha_2)y + \alpha_1\alpha_2)(y^{n-2} + ... + (-1)^n \alpha_3 ... \alpha_n), && (1) \\
b(x) &= t_{mn}(x^m + ... + (t_{1n}/t_{mn})x + (t_{0n}/t_{mn})) \\
&= (x^2 - (\beta_1 + \beta_2)x + \beta_1\beta_2)(t_{mn}x^{m-2} + ... + (-1)^m t_{mn}\beta_3 ... \beta_m), && (2) \\
c(y) &= t_{mn}(y^n + ... + (t_{m1}/t_{mn})y + (t_{m0}/t_{mn})) \\
&= (y^2 - (\gamma_1 + \gamma_2)y + \gamma_1\gamma_2)(t_{mn}y^{n-2} + ... + (-1)^n t_{mn}\gamma_3 ... \gamma_n), && (3) \\
d(x) &= t_{m0}(x^m + ... + (t_{10}/t_{m0})x + (t_{00}/t_{m0})) \\
&= t_{m0}(x^2 - (\delta_1 + \delta_2)x + \delta_1\delta_2)(x^{m-2} + ... + (-1)^m \delta_3 ... \delta_m), && (4)
\end{aligned}$$

then we have the relations :

$$\begin{aligned}
&(-1)^n \alpha_1\alpha_2 ... \alpha_n = t_{00}/t_{0n}, && (1') \\
&(-1)^m \beta_1\beta_2 ... \beta_m = t_{0n}/t_{mn}, && (2') \\
&(-1)^n \gamma_1\gamma_2 ... \gamma_n = t_{m0}/t_{mn}, \quad \text{and} && (3') \\
&(-1)^m \delta_1\delta_2 ... \delta_m = t_{00}/t_{m0}. && (4')
\end{aligned}$$

If we let

$$a_1(y) = \beta_1\beta_2 y^2 - \beta_1\beta_2(\alpha_1 + \alpha_2)y + \beta_1\beta_2\alpha_1\alpha_2,$$

and

$$a_2(y) = (-1)^m t_{mn}\beta_3 ... \beta_m y^{n-2} + ... + (-1)^{m+n} t_{mn}\beta_3 ... \beta_m \alpha_3 ... \alpha_n,$$

then by substituting formula (2') in equation (1), it will be true that

$$a(y) = a_1(y)a_2(y).$$

If

$$d_1(x) = \gamma_1\gamma_2 x^2 - \gamma_1\gamma_2(\delta_1 + \delta_2)x + \gamma_1\gamma_2\delta_1\delta_2,$$

and

$$d_2(y) = (-1)^n t_{mn}\gamma_3 ... \gamma_n x^{m-2} + ... + (-1)^{m+n} t_{mn}\gamma_3 ... \delta_n\delta_3 ... \delta_m,$$

then by substituting formula (3') into equation (4), it will be true that

$$d(x) = d_1(x)d_2(x).$$

Similarly, we can rewrite formulas (2) and (3) as

$$b(x) = b_1(x)b_2(x) \text{ and } c(y) = c_1(y)c_2(y).$$

To define the polynomial

$$p(x, y) = \textstyle\sum_{i=0}^{2}\sum_{j=0}^{2} p_{ij}x^i y^j,$$

let $p(1, 1) = 0$, p_{0i} be the coefficient of y^i in $a_1(y)$, p_{2i} be the coefficient of y^i in $c_1(y)$, p_{i0} be the coefficient of x^i in $d_1(x)$, and p_{i2} be the coefficient of x^i in $b_1(x)$.

To define the polynomial

$$q(x, y) = \textstyle\sum_{i=0}^{m-2}\sum_{j=0}^{n-2} q_{ij}x^i y^j,$$

let q_{0j} be the coefficient of y^j in $a_2(y)$, $q_{m-2,j}$ be the coefficient of y^j in $c_2(y)$, q_{i0} be the coefficient of x^i in $d_2(x)$, and $q_{i,n-2}$ be the coefficient of x^i in $b_2(x)$, and set all other coefficients equal to zero.

If we set

$$\begin{aligned} r(x, y) &= t(x, y) - p(x, y)q(x, y) \\ &= \sum_{i=0}^{m}\sum_{j=0}^{n} r_{ij}x^i y^j, \end{aligned}$$

then, it follows from the conditions on the roots that if r_{ij} is nonzero, then $1 \leq i \leq$

$m-1$ and $1 \leq j \leq n-1$.

The converse can be proved by observing that if

$$t(x,y) = p(x,y)q(x,y) + r(x,y),$$

where

$$r(x,y) = \textstyle\sum_{i=1}^{m-1} \sum_{j=1}^{m-1} r_{ij}x^i y^j,$$

then equations (1)-(4) can again be used to show that either

$$\alpha_1\alpha_2\delta_1\delta_2 = \beta_1\beta_2\gamma_1\gamma_2, \text{ and}$$

$$\alpha_3...\alpha_n\delta_3...\delta_m = \beta_3...\beta_m\gamma_3...\gamma_n$$

or

$$\alpha_1\alpha_2\beta_1\beta_2 = \gamma_1\gamma_2\delta_1\delta_2 \text{ and}$$

$$\alpha_3...\alpha_n\beta_3...\beta_m = \gamma_3...\gamma_n\delta_3...\delta_m.$$

Q.E.D.

COROLLARY 3.5. *If $t(x,y)$ is a polynomial defined by the formula*

$$t(x,y) = \sum_{i=0}^{4}\sum_{j=0}^{4} t_{ij}x^i y^j,$$

and $t(x,y)$ satifies the condition of Theorem 3.4, then there exist three polynomials $p(x,y)$, $q(x,y)$, and $r(x,y)$ such that

$$t(x,y) = p(x,y)q(x,y) + r(x,y),$$

where

$$p(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2} p_{ij}x^i y^j, \quad q(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2} q_{ij}x^i y^j,$$

and

$$r(x,y) = \sum_{i=1}^{3}\sum_{j=1}^{3} r_{ij}x^i y^j.$$

3.3. Decomposition Methods for Symmetric Operators

Most operators used in image processing are symmetric or skew-symmetric with respect to the x or y axis. The following section deals primarily with operators exhibiting some kind of symmetry.

DEFINITION 3.6. *A polynomial* $p(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x + a_0$, *is said to satisfy the symmetric property with respect to* n, *if* $a_i = a_{n-i}$ *for all* $i = 0, ..., n$.

Remark 3.7. We do not necessarily assume that the degree of $p(x)$ is equal to n. For example, if $p(x) = x$, then $p(x)$ satisfies the symmetric property with respect to the integer 2 .

DEFINITION 3.8. *A polynomial* $p(x)$ *is said to satisfy the skew symmetric property with respect to* n, *if* $a_i = -a_{n-i}$ *for all* $i = 0, ..., n$.

Remark 3.9. If $p(x)$ satisfies the symmetric property and has even degree, then the exponents of x^i, and x^{n-i} are of the same parity. Therefore, if $x = 0$ is a root of multiplicity k, then $p(x) = x^k p_1(x)$, where $p_1(x)$ is of even degree.

PROPOSITION 3.10. *If $p(x)$ is a polynomial satisfying the symmetric, or the skew symmetric property with respect to some integer n, then a non-zero number α is a root of $p(x)$ if and only if $1/\alpha$ is a root of $p(x)$.*

PROOF:

Since $p(1/\alpha) = (\pm 1/\alpha^n)p(\alpha)$ and α is a non zero root, the polynomial $p(\alpha) = 0$ if and only if the polynomial $p(1/\alpha) = 0$.

Q.E.D.

PROPOSITION 3.11. *If $p(x)$ is a polynomial of even degree n satisfying the symmetric property, then there exist two numbers k_1 and k_2 such that:*

$$p(x) = ax^{k_1}q_1(x)q_2(x)...q_{k_2}(x),$$

where a is the leading coefficient of $p(x)$, and each $q_j(x)$ is of the form $x^2 + b_jx + 1$.

PROOF:

If a is the coefficient of the highest degree of x in $p(x)$, and k_1 is the largest integer such that x^{k_1} divides $p(x)$, then $p(x) = ax^{k_1}p_1(x)$, where $p_1(x)$ satisfies the symmetric property. Since $p_1(x)$ is symmetric, and of even degree, we know by Proposition 3.10 that each root of $p_1(x)$ can be paired with its reciprocal. Thus, $p_1(x)$ has $n/2$ factors of the form $(x + \alpha)(x + 1/\alpha) = x^2 + (\alpha + 1/\alpha)x + 1$. Thus,

$$p_1(x) = (x^2 + b_1x + 1)...(x^2 + b_{(n-k_2)/2}x + 1).$$

Q.E.D.

PROPOSITION 3.12. *If $p(x) = \sum_{i=0}^{n} a_ix^i$ is a polynomial satisfying the skew symmetric property with respect to an even integer n, then there exists a polynomial $p_1(x)$ such that $p(x) = (x^2 - 1)p_1(x)$, where $p_1(x)$ satisfies the symmetric property.*

PROOF:

Since $a_i = -a_{n-i}$, where n is even,

$$p(1) = \sum_{i=0}^{n} a_i = \sum_{i=0}^{n/2} a_i - \sum_{i=0}^{n/2} a_i = 0.$$

Similarly, $p(-1) = 0$. Thus, $x^2 - 1$ is a factor of $p(x)$.

We must now show that if $p(x) = (x^2 - 1)p_1(x)$, then $p_1(x)$ satisfies the symmetric property. If

$$p_1(x) = b_{n-2}x^{n-2} + \ldots + b_1 x + b_0,$$

then note that

$$a_0 = -b_0, a_1 = -b_1, a_{n-1} = b_{n-3}, \text{ and } a_n = b_{n-2}.$$

Since $p(x)$ satisfies the skew symmetric property, $b_{n-2} = b_0$, and $b_{n-3} = b_1$. Note also that $a_k = b_{k-2} - b_k$, for all $k = 2, \ldots, n-2$.

We must now show that $b_k = b_{n-2-k}$. If we assume inductively that $b_{k-2} = b_{n-k}$, then, since $p(x)$ satisfies the skew symmetric property,

$$b_{k-2} - b_k = a_k = -a_{n-k} = -(b_{n-k-2} - b_{n-k}),$$

so that $b_k = b_{n-2-k}$. Thus, $p_1(x)$ is symmetric.

Q.E.D.

COROLLARY 3.13. *If $p(x)$ is a polynomial satisfying the skew symmetric property with respect to an even integer n, then there exist two numbers k_1 and k_2 such that*

$$p(x) = a(x^2 - 1)x^{k_1} q_1(x) q_2(x) \ldots q_{k_2}(x),$$

where a is the leading coefficient of $p(x)$, and $q_j(x)$ is of the form $(x^2 + b_j x + 1)$, for $j = 1, \ldots, k_2$.

PROOF:

If zero is a root of $p(x)$, then let k_1 be its multiplicity. By Proposition 3.12 and Remark 3.9, there is a symmetric polynomial $p_1(x)$, such that

$$p(x) = a_n x^{k_1}(x^2 - 1)p_1(x).$$

Since n is even, and $p_1(x)$ satisfies the symmetric property with respect to n, Proposition 3.11 implies that $p_1(x)$ factors into a product of quadratic polynomials of the form $x^2 + bx + 1$.

Q.E.D.

Remark 3.14. If

$$p(x,y) = \sum_{i=0}^{m}\sum_{j=0}^{n} t_{ij}x^i y^j,$$

then $p(x,y)$ can be written in the form

$$\sum_{i=0}^{m} x^i(\sum_{j=0}^{n} t_{ij}y^j) = \sum_{i=0}^{n} x^i p_i(y).$$

The polynomial $p(x,y)$ can also be written in the form

$$\sum_{j=0}^{m} y^j(\sum_{i=0}^{n} t_{ij}x^i) = \sum_{j=0}^{m} y^j q_j(x).$$

DEFINITION 3.15. *If $p(x,y) = \sum_{i=0}^{m}\sum_{j=0}^{n} t_{ij}x^i y^j = \sum_{i=0}^{n} x^i p_i(y)$, then the polynomial $p(x,y)$ is said to satisfy the symmetric (respectively the skew symmetric) property with respect to y, if $p_i(y)$ satisfies the symmetric (respectively the skew symmetric) property, for all $i = 0, ..., n$.*

A similar definition can be given for the variable x.

PROPOSITION 3.16. *If $p(x,y)$ is any polynomial, then*

$$p(x,y) = p_1(x,y) + p_2(x,y),$$

where $p_1(x,y)$ satisfies the symmetric property with respect to y, and $p_2(x,y)$ satisfies the skew symmetric property with respect to y.

PROOF:

Let $p_1(y) = \sum_{j=0}^{n} \frac{a_j + a_{n-j}}{2} y^j$ and $p_2(y) = \sum_{j=0}^{n} \frac{a_j - a_{n-j}}{2} y^j$.

Q.E.D.

A similar proposition can be given for the variable x.

PROPOSITION 3.17. *If $t(x,y) = \sum_{i=0}^{m} \sum_{j=0}^{n} t_{ij} x^i y^j$ has the property that $t_{00} = t_{0n} = t_{m0} = t_{mn} = 0$; and $t_{10} = kt_{01}$, $t_{m,n-1} = kt_{m-1,n}$, $t_{m-1,0} = kt_{m1}$, and $t_{0,n-1} = kt_{1n}$, where k = +1 or -1, then there exist three polynomials $p(x,y)$, $q(x,y)$, and $r(x,y)$ such that :*

$$t(x,y) = p(x,y)q(x,y) + r(x,y),$$

where

$$p(x,y) = \sum_{i=0}^{2} \sum_{j=0}^{2} p_{ij} x^i y^j, \quad q(x,y) = \sum_{i=0}^{m-2} \sum_{j=0}^{n-2} q_{ij} x^i y^j,$$

and

$$r(x,y) = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} r_{ij} x^i y^j.$$

PROOF:

If k = 1, let

$$p(x,y) = x + y + xy^2 + x^2y,$$

$$q(x,y) = [t_{01} + ... + t_{0,n-1}y^{n-2}] + [t_{1n}y^{n-2} + ... + t_{m-1,n}x^{m-2}y^{n-2}]$$
$$+ [t_{m1}x^{m-2} + ... + t_{m,n-2}x^{m-2}y^{n-3}] + [t_{20}x + ... + t_{m-2,0}x^{m-3}],$$

and

$$r(x,y) = t(x,y) - p(x,y)q(x,y).$$

If k = -1, let $p(x,y) = x - y - xy^2 + x^2y$, $q(x,y)$, and $r(x,y)$ as above.

Q.E.D.

Theorem 3.1 has the foolowing corollaries

COROLLARY 1 TO THEOREM 3.1. *If*

$$t(x,y) = \sum_{i=0}^{m}\sum_{j=0}^{n} t_{ij}x^iy^j$$

satisfies the symmetric or the skew symmetric property with respect to y, *with* t_{00} *and* t_{m0} *nonzero, then there exist three polynomials* $p(x,y)$, $q(x,y)$, *and* $r(x,y)$ *such that*

$$t(x,y) = p(x,y)q(x,y) + r(x,y),$$

where

$$p(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2} p_{ij}x^iy^j, \quad q(x,y) = \sum_{i=0}^{m-2}\sum_{j=0}^{n-2} q_{ij}x^iy^j,$$

and

$$r(x,y) = \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} r_{ij}x^iy^j.$$

PROOF:

Note that since $t(x,y)$ is either symmetric, or skew symmetric with respect to y, t_{0n}, and t_{mn} are also both nonzero. Following the notation used in the proof of Theorem 3.1, the polynomials $a(y)$, $b(x)$, $c(y)$, and $d(x)$ are such that $b(x) = d(x)$, and $a(y)$ and $c(y)$ satisfy the symmetric property. Therefore, by Proposition 3.11 or Corollary 3.13, each root of $a(y)$ and $c(y)$ can be paired with its reciprocal, and we can choose $a_1(y), a_2(y), c_1(y)$, and $c_2(y)$ such that $\alpha_1\alpha_2 = 1$, $\alpha_3\alpha_4...\alpha_n = 1$, $\gamma_1\gamma_2 = 1$, and $\gamma_3\gamma_4...\gamma_n = 1$. Thus, the conditions of Theorem 3.1 are satisfied, and the result follows.

Q.E.D.

COROLLARY 2 TO THEOREM 3.1. *If*

$$t(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{n} t_{ij}x^iy^j$$

satisfies the symmetric property with respect to both x and y and if the matrix $T = (t_{ij})$ is symmetric, then there exist three polynomials $p(x,y)$, $q(x,y)$ and $r(x,y)$ such that :

$$t(x,y) = p(x,y)q(x,y) + r(x,y),$$

where

$$p(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2} p_{ij}x^iy^j, \quad q(x,y) = \sum_{i=0}^{n-2}\sum_{j=0}^{n-2} q_{ij}x^iy^j,$$

and

$$r(x,y) = \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} r_{ij}x^iy^j.$$

Moreover, the polynomials $p(x,y), q(x,y)$, and $r(x,y)$ can also be chosen to be symmetric with respect to both x and y.

PROOF:

In the case where $t_{00} \neq 0$, the polynomials $a(y)$, $b(x)$, $c(y)$ and $d(x)$ defined in Theorem 3.1 all have the same coefficients and therefore have the same decompositions. In the case, where $t_{00} = t_{0n} = t_{m0} = t_{mn} = 0$, the result follows from Proposition 3.17.

Q.E.D.

3.4. The LU Factorization

For a detailed discussion of how the LU Factorization Theorem can be used to implement operators in such a way that their operation count is reduced, see Nikias et. al [21].

PROPOSITION 3.18. *If a 5×5 matrix* $\mathbf{M}$ *has an LU factorization (without permutations), then there exist two rank one matrices A and B and a matrix $R = (r_{ij})$ such that*

$$\mathbf{M} = A + B + R,$$

and $r_{1j} = r_{2j} = r_{i1} = r_{i2} = 0$, *for* $i,j = 1, \ldots 5$.

PROOF:

Suppose $\mathbf{M}$ can be written $\mathbf{M} = L \cdot U$, where L is an lower triangular matrix, and U is an upper triangular matrix. Suppose $L = (l_{ij})$, and $U = (u_{ij})$.

Let U_i be the 5×5 zero matrix whose i^{th} row is the i^{th} row of U and let $V = U_3 + U_4 + U_5$. M can then be rewritten as

$$M = L \cdot U = L \cdot (U_1 + U_2 + V) = L \cdot U_1 + L \cdot U_2 + L \cdot V = A + B + R,$$

where

$$A = L \cdot U_1,\ B = L \cdot U_2, \text{ and } R = L \cdot V.$$

The matrix A is a rank one matrix of the form

$$A = \begin{bmatrix} l_{11} \\ l_{21} \\ l_{31} \\ l_{41} \\ l_{51} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} & u_{15} \end{bmatrix}$$

and the matrix B is a rank one matrix of the form

$$B = \begin{bmatrix} 0 \\ l_{22} \\ l_{32} \\ l_{42} \\ l_{52} \end{bmatrix} \begin{bmatrix} 0 & 1 & u_{23} & u_{24} & u_{25} \end{bmatrix}.$$

On the other hand, R is a matrix whose first two rows and columns are all zero.

Q.E.D.

Thus, if $\mathbf{t}$ is a 5×5 operator, it can be viewed as a 5×5 matrix. If $\mathbf{t}$ has an LU factorization without permutations, then there exist two rank one operators $\mathbf{u}$, and $\mathbf{v}$ and an operator $\mathbf{r}$ such that $\mathbf{t} = \mathbf{u} + \mathbf{v} + \mathbf{r}$.

Since rank one operators are separable (i.e. have an exact factorization), there exist four 3×3 operators $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \mathbf{t}_4$, such that

$$\mathbf{u} = \mathbf{t}_1 \oplus \mathbf{t}_2 \quad \text{and} \quad \mathbf{v} = \mathbf{t}_3 \oplus \mathbf{t}_4.$$

Therefore, $\mathbf{t}$ has the following decomposition

$$\mathbf{t} = (\mathbf{t}_1 \oplus \mathbf{t}_2) + (\mathbf{t}_3 \oplus \mathbf{t}_4) + \mathbf{r},$$

where

$$\mathbf{r} = \begin{bmatrix} < t_{11} > & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}.$$

While Proposition 3.18 appears to be as good as the results in Corollary 3.2, note that the center pixel in the image is at the (1,1) location. Thus, the data must be shifted before it can be convolved with the operator $\mathbf{r}$.

While the methods discussed in Sections 1-4 give techniques for the decomposition of an $n \times n$ operator, most operators used in image processing are either symmetric or skew symmetric with respect to either the x or the y axis. If n is an odd integer, then the rank of the operator is no more than $(n+1)/2$ in the symmetric case, and no more than $(n-1)/2$ in the skew symmetric case. Using the rank method or the LU factorization on symmetric or skew symmetric operators of low rank, it is frequently the case that a decomposition can be produced which will be nearly as efficient an implementation as that provided by our methods. Thus, the methods presented here, are more apt to provide a decomposition superior to those provided by the LU factorization on templates of relatively full rank.

3.5. Gröbner Bases

Gröbner bases have been introduced to Bruno Buchberger [4] in his dissertation in 1965. They are named after W. Gröbner [9], his thesis advisor. Buchberger's

method of Gröbner bases is a technique that provides algorithmic solutions to a variety of problems in ideal theory as well as solutions to numerous problems in many other fields. It is a general purpose method for multivariate polynomial computations. It is implemented in all major computer algebra systems such as MACSYMA, MAPLE, MuMATH, REDUCE, SCRATCHPAD II, etc... In spite of its simplicity, the Buchberger algorithm solves a wide range of problems from symbolic algebra to computational geometry.

In this dissertation, we will apply it to systems of algebraic equations and linear homogeneous equations with polynomial coefficients.

3.5.1. Terminology

Let $\mathbf{F}$ be a field, and let $\mathbf{F}[y_1, y_2, ..., y_n]$ be the ring of n-variate polynomials over $\mathbf{F}$. The letters f, g, h will stand as typed variables for polynomials in $\mathbf{F}[y_1, y_2, ..., y_n]$; H, G for finite subsets of $\mathbf{F}[y_1, y_2, ..., y_n]$; t, u for power products of the form $y_1^{i_1}...y_n^{i_n}$; a,b for elements in the field $\mathbf{F}$, and i, j for natural numbers.

Definition 3.19. *If H is defined as above, then the ideal generated by H is the set*

$$Ideal(H) = \{\sum f_i h_i : h_i \in H, f_i \in \mathbf{F}[y_1, y_2, ..., y_n]\}$$

of all linear combinations of the elements of H in $\mathbf{F}[y_1, y_2, ..., y_n]$.

Definition 3.20. *A total ordering $<$ on the power products $y_1^{i_1}...y_n^{i_n}$ is called admissible if $1 = y_1^0...y_n^0$ is minimal under $<$ and multiplication by a power product preserves the ordering.*

Examples for such orderings are the total degree ordering (i.e. $1 < y_1 < y_2 < y_1^2 < y_1y_2 < y_2^2 < y_1^3 < \ldots$ in the bivariate case), and the purely lexicographic ordering (i.e. $1 < y_1 < y_1^2 < \ldots < y_2 < y_1y_2 < y_1^2y_2 < \ldots < y_2^2 < y_1y_2^2 < \ldots$ in the bivariate case).

Given a fixed ordering $<$, we will denote the coefficient of t in g by $\mathrm{coeff}(g, t)$, the leading power product of g with respect to $<$ by $\mathrm{lpp}(g)$, and the leading coefficient of g with respect to $<$ by $\mathrm{lc}(g)$.

DEFINITION 3.21. *Given a nonzero polynomial f, and two other polynomials g, and h in $\mathbf{F}[y_1, y_2, \ldots, y_n]$, we say that g reduces to h modulo f or $g \rightarrow_h f$, if and only if there exist u, b, $b \neq 0$, such that $b.u.lpp(h)$ is identical with some monomial of g, and $f = g - b.u.h$.*

Such a reduction can be viewed as a generalized division, deleting a monomial in g. The result f is always strictly smaller than g, with respect to the ordering $<$.

Example 1. Let $\mathbf{F} = \mathbf{Q}$, $g = y_1y_2y_3 - y_1^2$, and $h = y_1y_3 - y_2$. If we let $u = y_2$, and $b = 1$, and we use the lexicographic ordering, then $f = g - b.u.h = y_2^2 - y_1^2$.

DEFINITION 3.22. *We say that f is in normal form (or reduced form) if and only if there is no f' such that $f \rightarrow_h f'$.*

Example 2. Let g and h be as in Example 1. If $h_1 = h$, and $h_2 = y_1y_2 - y_1$, then $f = y_2^2 - y_1^2$ of Example 1 is irreducible modulo $\{h_1, h_2\}$.

Another reduction of g is the following one: $g \rightarrow f' = y_1y_3 - y_1^2 \rightarrow f" = y_2 - y_1^2$. In this case, $h"$ is irreducible modulo $\{h_1, h_2\}$.

Therefore, both f and f'' are normal forms of g modulo $\{h_1, h_2\}$.

Hence, there exist several reduction paths that can lead to a normal form, modulo a basis (or a set) H.

DEFINITION 3.23. *The set H is called a Gröbner basis if and only if each g has a unique normal form modulo H.*

We will note that different orderings will give rise to different Gröbner bases, and that a Gröbner basis is not necessarily unique. It is possible to reduce a polynomial of the basis modulo another polynomial of the basis, and hence obtain a smaller basis.

As a matter of experience, using total degree ordering yields much better computing time in general. On the other hand, when using the purely lexicographic ordering, the resulting basis is in triangular form (i.e. each polynomial introduces at most one new variable).

3.5.2. Application to Systems of Equations and Algorithm

Given a system of algebraic equations H, the Gröbner basis can be used to:

-decide whether H is solvable;

-decide whether H has finitely or infinitely many solutions;

-find all solutions of H, in the finite case.

The following algorithm shows how the Gröbner basis method using the lexicographic ordering can be applied to determine the answer to a system of equations.

<u>Algorithm</u>

G = Gröbner basis of H.

If $1 \in G$

then H is unsolvable.

Else if there exist an integer j such that no leading power product of the polynomials in G is of the form y_j^e, for some e,

then H has infinitely many solutions,

else do f= the (only) polynomial in $G \cap F[y_1]$,

$X_1 = \{(a) : f(a) = 0\}$,

for $i = 1, ..., n-1$,

$X_{i+1} = \emptyset$,

for all $(a_1, ..., a_i) \in X_i$, do

$K = \{g(a_1, ..., a_i, y_{i+1}) : g \in G \cap F[y_1, ..., y_{i+1}]$
$\backslash F[y_1, ..., y_i]\}$,

f = g.c.d. of the polynomials in K,

$X_{i+1} = X_i \cup \{(a_1, ...a_i, a) : f(a) = 0\}$.

Finally, X_n contains all the solutions of H.

The Gröbner basis method was used in some examples of Chapter 5 to show that a given template cannot be decomposed into a certain form. Several examples are presented in Chapter 5. The equations were set up as a system which contains both linear and nonlinear equations. For example, for a 5×5 template, we will have 25 equations, 17 of which are not linear. To avoid having to use Gröbner bases theory to solve a system of nonlinear equations, we reduced the problem to roots of polynomials of one variable to decompose the boundary of the template. Once that outer boundary has been matched we only face a system of linear equations to

match the second boundary. While the Gröbner bases method can be very slow in solving nonlinear equations, it does a very fine job on solving linear equations. All computations were done on MAPLE.

CHAPTER 4

VARIANT OPERATORS

We now turn our attention from results concerning shift-invariant templates to analogous results for templates, which are not necessarily shift-invariant. Recall that the Discrete Fourier Transform and the Gabor Transform are examples of variant templates. We will try to determine the class of templates that are separable (i.e. decomposable into the convolution of a horizontal template and a vertical template or vice-versa). Since not all templates are separable, we will show that every template can be written as the sum of separable templates.

4.1. Decomposition of General Operators

DEFINITION 3.1. *A template* $\mathbf{t}$ *defined on* $\mathbf{Z} \times \mathbf{Z}$ *has column rank one if it has finite support and there exists a column vector* $\mathbf{u}$ *such that for each pixel location* $\mathbf{x}$ *in* $\mathbf{Z} \times \mathbf{Z}$ *and every column vector* $\mathbf{v}$ *in* $\mathbf{t}(\mathbf{x})$*, there exists a scalar* λ *such that* $\mathbf{v} = \lambda\mathbf{u}$.

A similar definition could be given for row rank one templates.

DEFINITION 4.2. *A template* $\mathbf{t}$ *defined on* $\mathbf{Z} \times \mathbf{Z}$ *and having finite support is said to be horizontal if, for every* $\mathbf{x}$ *in* $\mathbf{Z} \times \mathbf{Z}$, *there exists a unique nonzero row vector* $\mathbf{v}$

in $\mathbf{t}(\mathbf{x})$. *Similarly, a template* $\mathbf{t}$ *is said to be vertical if, for every* $\mathbf{x}$ *in* $\mathbf{Z} \times \mathbf{Z}$, *there exists a unique nonzero column vector* $\mathbf{v}$ *in* $\mathbf{t}(\mathbf{x})$.

THEOREM 4.3. *If* $\mathbf{t}$ *is a column rank one template defined on* $\mathbf{Z} \times \mathbf{Z}$ *and has finite support, then there exist an invariant vertical template* $\mathbf{t}_1$ *and a (possibly variant) horizontal template* $\mathbf{t}_2$ *such that* $\mathbf{t} = \mathbf{t}_1 \oplus \mathbf{t}_2$.

PROOF:

For every $\mathbf{x}$ in $\mathbf{Z} \times \mathbf{Z}$, $\mathbf{t}(\mathbf{x})$ is an $m \times n$ array. This array contains n column vectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n$. Since the template has rank one, for each $i = 1, ..., n$, there exists a scalar λ_i and a fixed column vector $\mathbf{u}$ such that $\mathbf{v}_i = \lambda_i \mathbf{u}$. Therefore,

$$\mathbf{t}(\mathbf{x}) = [\mathbf{u}] \bullet [\lambda_1 \;\; \lambda_2 \;\; ... \;\; \lambda_n] \quad \text{(outer product)},$$

where $[\mathbf{u}]$ is an invariant vertical template and $[\lambda_1 \;\; \lambda_2 \;\; ... \;\; \lambda_n]$ is a (possibly variant) horizontal template.

Q.E.D.

Remark 4.4. Let $\mathbf{X}$ be an $n \times n$ subset of $\mathbf{Z} \times \mathbf{Z}$, and let $\mathbf{F}$ be a value set. The two-dimensional Fourier template $\mathbf{K}$ is defined by

$$\mathbf{K}(u, v) = \frac{1}{n} exp\{-(uX_1 + vY_1)2\pi i/n\},$$

where $X_1, Y_1 \in \mathbf{F}^{\mathbf{X}}$ are defined by

$$X_1 = \{((x,y),z) : x = z\} \text{ and } Y_1 = \{((x,y),z) : y = z\}.$$

Even though it can be shown that $\mathbf{K}$ is a separable variant template, (i.e. it splits into the product of a horizontal and a vertical), $\mathbf{K}$ has neither row rank one

nor column rank one. If the Fourier template $\mathbf{K}$ did have column rank one, then by Theorem 4.3, it could be factored into the product of an invariant vertical template and a variant horizontal template.

DEFINITION 4.5. *If a template* $\mathbf{t}$ *defined on* $\mathbf{Z} \times \mathbf{Z}$ *has finite support, then we say that* $\mathbf{t}$ *has column rank at most* k *if there exist* k *column vectors* $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k$ *such that for every pixel location* $\mathbf{x}$ *in* $\mathbf{Z} \times \mathbf{Z}$ *and every column vector* $\mathbf{v}$ *in* $\mathbf{t}(\mathbf{x})$*, there exist* k *scalars* $\lambda_1, \lambda_2, ..., \lambda_k$ *such that* $\mathbf{v} = \sum_{i=1}^{k} \lambda_i \mathbf{u}_i$.

Q.E.D.

THEOREM 4.6. *If a column rank* k *template* $\mathbf{t}$ *defined on* $\mathbf{Z} \times \mathbf{Z}$ *has finite support, then there exist* k *column rank one (possibly variant) templates* $\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_k$ *such that* $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2 + ... + \mathbf{t}_k$.

PROOF:

By Definition 4.5, for every pixel location $\mathbf{x}$ in $\mathbf{Z} \times \mathbf{Z}$, and every column vector $\mathbf{v}_i$ in $\mathbf{t}(\mathbf{x})$, there exist k scalars $\lambda_{i1}, \lambda_{i2}, ..., \lambda_{ik}$ such that $\mathbf{v}_i = \sum_{j=1}^{k} \lambda_{ij} \mathbf{u}_j$.

Let $\mathbf{v}_{ij} = \lambda_{ij} \mathbf{u}_j$ for $i = 1, ..., n$. If $\mathbf{t}_j(x) = [\mathbf{v}_{1j} \;\; \mathbf{v}_{2j} \;\; \cdots \;\; \mathbf{v}_{nj}]$ for $j = 1, ..., k$, then each $\mathbf{t}_j$ is a column rank one template and $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2 + ... + \mathbf{t}_k$.

COROLLARY 4.7. *If a template* $\mathbf{t}$ *defined on* $\mathbf{Z} \times \mathbf{Z}$ *has finite support and column rank* k*, then there exist* k *vertical invariant templates* $\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_k$*, and* k *horizontal variant templates* $\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_k$*, such that*

$$\mathbf{t} = \sum_{i=1}^{k} \mathbf{r}_i \oplus \mathbf{s}_i.$$

PROOF:

This result follows immediately from Theorems 4.3 and 4.6.

Q.E.D.

Note that a result similar to Corollary 4.7 can be proved for row rank k templates.

THEOREM 4.8. *Let* $\mathbf{t}$ *be a template on a finite rectangular subset of the plane with* m *rows and* n *columns, and let* $M_{\mathbf{t}}$ *be the matrix described in section 4 of Chapter 2. If* A_{rs} *denotes the matrix* $[m_{n(r-1)+i,n(j-1)+s}]$ *for* $i = 1, ..., n$ *and* $j = 1, ..., n$, *then the template* $\mathbf{t}$ *is separable if and only if each matrix* A_{rs} *has rank one for all* $r = 1, ..., m$ *and* $s = 1, ..., m$.

PROOF:

Since for every $r = 1, ..., m$ and $s = 1, ..., m$, A_{rs} is an $n \times n$ rank one matrix, there exist a column vector $\mathbf{v}_{rs}$ and a row vector $\mathbf{h}_{rs}$ such that $M_{rs} = \mathbf{v}_{rs}\mathbf{h}_{rs}$.

Let the column vectors $\mathbf{v}_{1s}, \mathbf{v}_{2s}, ..., \mathbf{v}_{ns}$ form the s^{th} $n \times n$ block of the block diagonal matrix V, for $s = 1, ..., m$.

On the other hand, let the i_{th} elements of the n row vectors $\mathbf{h}_{r1}, \mathbf{h}_{r2}, ..., \mathbf{h}_{rn}$ form the diagonal elements of the (r, i) diagonal block of the block matrix H, for $r = 1, ..., m$.

A direct computation $V \cdot H$ will give the matrix $M_{\mathbf{t}}$. Under the isomorphism ψ, V and H represent the image of a vertical and a horizontal template, respectively. We then conclude by Theorem 2.6, that $\mathbf{t}$ is separable.

Q.E.D.

COROLLARY 4.9. *Let* $\mathbf{t}$ *be a template on a finite rectangular subset of the plane with* m *rows and* n *columns, and let* $M_{\mathbf{t}}$ *be the matrix described in the previous paragraph. If* A_{rs} *denotes the matrix* $[m_{n(r-1)+i,n(j-1)+s}]$ *for* $i = 1, ..., n$ *and* $j = 1, ..., n$, *then the template* $\mathbf{t}$ *can be written as the sum of* k *separable templates if and only if each matrix* A_{rs} *has rank at most* k *for all* $r = 1, ..., m$ *and* $s = 1, ..., m$.

PROOF:

Since for every $r = 1, ..., m$ and $s = 1, ..., m$, A_{rs} is an $n \times n$ matrix of at most rank k, there exist k column vectors $\mathbf{v}^1_{rs}, \mathbf{v}^2_{rs}, ..., \mathbf{v}^k_{rs}$ and k row vectors $\mathbf{h}^1_{rs}, \mathbf{h}^2_{rs}, ..., \mathbf{h}^k_{rs}$ such that if $A^j_{rs} = \mathbf{v}^j_{rs}\mathbf{h}^j_{rs}$, for $j = 1, ..., k$, then $A_{rs} = A^1_{rs} + A^2_{rs} + ... + A^k_{rs}$. For $j = 1, ..., k$, A^j_{rs} is an $n \times n$ rank one matrix, and therefore Theorem 4.9 applies. If for every $j = 1, ..., k$ M_j is the block matrix $[A^j_{rs}]$, for $r = 1, ..., m$ and $s = 1, ..., m$, then there exists two matrices V_j and H_j such that $A^j_{rs} = V_j \cdot H_j$.

Since $M_{\mathbf{t}} = M_1 + M_2 + ... + M_k$ implies that $M_{\mathbf{t}} = V_1 \cdot H_1 + V_2 \cdot H_2 + ... + V_k \cdot H_k$, Theorem 4.8 can now be used to conclude that the template $\mathbf{t}$ is the sum of k separable templates.

Q.E.D.

Remark 4.10. Similarly, if we let A_{rs} denote the matrix $[m_{n(i-1)+r,n(s-1)+j}]$, then we can expect the same type of results as in Theorem 4.8. The only difference is that the horizontal and the vertical templates are convolved in the reverse order. Recall also that the two-dimensional Fourier template is separable. It is straightforward to check that the Fourier template has the property that each of the matrices A_{rs} has rank one.

Remark 4.11. Note that according to the definition of the convolution of two tem-

plates $\mathbf{t} \oplus \mathbf{s}$, it is as if we are "sliding" the first template $\mathbf{t}$ along the second one $\mathbf{s}$. Therefore, in order to minimize the number of equations and therefore minimize the complexity of the solution, it will be very "helpful" to have the first template $\mathbf{t}$ invariant. The goal of the next five theorems is to make this last statement more precise. In fact, in this case, polynomials (rather than template notation) can be used in the decomposition.

THEOREM 4.12. *If* $\mathbf{t}$ *is a* 5×5 *variant template, then there exist seven templates* $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \mathbf{t}_4, \mathbf{t}_5, \mathbf{t}_6,$ *and* $\mathbf{t}_7$ *such that*

$$\mathbf{t} = (\mathbf{t}_1 \oplus \mathbf{t}_2) + (\mathbf{t}_3 \oplus \mathbf{t}_4) + (\mathbf{t}_5 \oplus \mathbf{t}_6) + \mathbf{t}_7,$$

where $\mathbf{t}_1, \mathbf{t}_3,$ *and* $\mathbf{t}_5$ *are* 3×3 *invariant templates and* $\mathbf{t}_2, \mathbf{t}_4, \mathbf{t}_6,$ *and* $\mathbf{t}_7$ *are* 3×3 *variant templates.*

PROOF:

Let $\mathbf{t}$ be of the form

$$\mathbf{t} = \begin{bmatrix} t_{-2,-2} & t_{-2,-1} & t_{-2,0} & t_{-2,1} & t_{-2,2} \\ t_{-1,-2} & t_{-1,-1} & t_{-1,0} & t_{-1,1} & t_{-1,2} \\ t_{0,-2} & t_{0,-1} & < t_{0,0} > & t_{0,1} & t_{0,2} \\ t_{1,-2} & t_{1,-1} & t_{1,0} & t_{1,1} & t_{1,2} \\ t_{2,-2} & t_{2,-1} & t_{2,0} & t_{2,1} & t_{2,2} \end{bmatrix}$$

and let us define the following shift invariant templates:

$$\mathbf{t}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & <0> & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{t}_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & <0> & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \text{and } \mathbf{t}_5 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & <0> & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

In addition, let us define the following variant templates:

$$\mathbf{t}_2 = \begin{bmatrix} t_{-2,-2} & t_{-2,-1} & 0 \\ t_{-1,-2} & <0> & t_{1,2} \\ 0 & t_{2,1} & t_{2,2} \end{bmatrix}, \quad \mathbf{t}_4 = \begin{bmatrix} 0 & t_{-2,1} & t_{-2,2} \\ t_{1,-2} & <0> & t_{-1,2} \\ t_{2,-2} & t_{2,-1} & 0 \end{bmatrix},$$

and

$$\mathbf{t}_6 = \begin{bmatrix} 0 & t_{-2,0} & 0 \\ t_{0,-2} & <0> & t_{0,2} \\ 0 & t_{2,0} & 0 \end{bmatrix}.$$

Note that a routine calculation using the definition of the $\oplus$ operation can be used to show that the template $\mathbf{t}_7 = \mathbf{t} - (\mathbf{t}_1 \oplus \mathbf{t}_2 + \mathbf{t}_3 \oplus \mathbf{t}_4 + \mathbf{t}_5 \oplus \mathbf{t}_6)$ is a 3×3 template. Thus, $\mathbf{t}$ can be decomposed as the sum and product of at most seven 3×3 templates.

Q.E.D.

4.2. Decomposition of Symmetric Operators

THEOREM 4.13. *If* $\mathbf{t}$ *is a* $(2m+1) \times (2n+1)$ *variant template symmetric with respect to both axes, then there exist four templates* $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3,$ *and* $\mathbf{t}_4$*, such that*

$$\mathbf{t} = (\mathbf{t}_1 \oplus \mathbf{t}_2) + \mathbf{t}_3 + \mathbf{t}_4,$$

where $\mathbf{t}_1$ *is a 3* $\times$ *3 invariant template,* $\mathbf{t}_2$ *is an* $(2m-1) \times (2n-1)$ *variant templates symmetric with respect to both axes,* $\mathbf{t}_3$ *is a* $(2m+1) \times 1$ *vertical template, and* $\mathbf{t}_4$ *is a* $1 \times (2n+1)$ *horizontal template.*

PROOF:

Let $\mathbf{t}_2$ be a $(2m-1) \times (2n-1)$ template symmetric with respect to both axes, whose left top $m \times n$ corner is given by the equations:

$$\mathbf{t}_{2_{-m+i,-n+j}} = \begin{cases} \mathbf{t}_{-m+i,-n+j} & \text{if} \quad (i,j) = (0,0),(0,1),(1,0),(1,1) \\ \mathbf{r}(i,j) & \text{otherwise,} \end{cases},$$

where

$$\mathbf{r}(i,j) = \begin{cases} \mathbf{t}_{-m+i,-n+j} + \mathbf{t}_{-m+i,-n+j-2} & \text{if} \quad i < j \\ \mathbf{t}_{-m+i,-n+j} + \mathbf{t}_{-m+i-2,-n+j} & \text{if} \quad i > j \\ \mathbf{t}_{-m+i,-n+j} + \mathbf{t}_{-m+i-2,-n+j} & \\ +t_{-m+i,-n+j-2} + \mathbf{t}_{-m+i-2,-n+j-2} & \text{if} \quad i = j. \end{cases}$$

Because of the symmetry of $\mathbf{t}_2$, it is easy to know all the other entries of that template.

Now if $\mathbf{t}_1$ is the template $\mathbf{t}_1 = \begin{bmatrix} 1 & 0 & -1 \\ 0 & <0> & 0 \\ -1 & 0 & 1 \end{bmatrix}$, then $\mathbf{t} - (\mathbf{t}_1 \oplus \mathbf{t}_2)$ is simply the sum of a vertical template $\mathbf{t}_3$ and a horizontal template $\mathbf{t}_4$.

Q.E.D.

THEOREM 4.14. *If* $\mathbf{t}$ *is a* $(2m+1) \times (2n+1)$ *variant template symmetric with respect to both axes, then there exist five templates* $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \mathbf{t}_4$, *and* $\mathbf{t}_5$ *such that*

$$\mathbf{t} = (\mathbf{t}_1 \oplus \mathbf{t}_2) + (\mathbf{t}_3 \oplus \mathbf{t}_4) + \mathbf{t}_5,$$

where $\mathbf{t}_1$ *and* $\mathbf{t}_3$ *are 3 × 3 invariant templates and* $\mathbf{t}_2, \mathbf{t}_4$, *and* $\mathbf{t}_5$ *are* $(2m-1) \times (2n-1)$ *variant templates symmetric with respect to both axes.*

PROOF:

The main idea in this proof is to decompose the cruciform template $\mathbf{s}$ (i.e. $\mathbf{s}_{\mathbf{ij}} = 0$, whenever $i \neq 0$ or $j \neq 0$) derived from the proof of Theorem 4.14.

If

$$\mathbf{t}_3 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & <0> & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

and $\mathbf{t}_4$ is the cruciform template of size $(2m-1) \times (2n-1)$ with same extremal elements as in $\mathbf{s}$, then a simple computation shows that $\mathbf{t} - (\mathbf{t}_1 \oplus \mathbf{t}_2 + \mathbf{t}_3 \oplus \mathbf{t}_4)$ is a

$(2m+1) \times (2n+1)$ template symmetric with respect to both axes.

Q.E.D.

THEOREM 4.15. *If* $\mathbf{t}$ *is a* $(2m+1) \times (2n+1)$ *variant template skew-symmetric with respect to both the vertical and the horizontal axes, then there exist two templates* $\mathbf{t}_1$ *and* $\mathbf{t}_2$ *such that*

$$\mathbf{t} = \mathbf{t}_1 \oplus \mathbf{t}_2,$$

where $\mathbf{t}_1$ *is the invariant template defined by*

$$\mathbf{t}_1 = \begin{bmatrix} 1 & 0 & -1 \\ 0 & <0> & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

and $\mathbf{t}_2$ *is a* $(2m-1) \times (2n-1)$ *variant template symmetric with respect to both axes.*

PROOF:

If $\mathbf{t}_2 = [\mathbf{t}_{2ij}]$ is the $(2m-1) \times (2n-1)$ template symmetric with respect to both axes defined in the proof of Theorem 4.13, then it is a matter of computation to check that $\mathbf{t}_1 \oplus \mathbf{t}_2$ is the given $(2m+1) \times (2n+1)$ skew-symmetric template $\mathbf{t}$.

Q.E.D.

THEOREM 4.16. *If* $\mathbf{t}$ *is a* $(2m+1) \times (2m+1)$ *totally symmetric template and if the corner values are zero, then there exist three templates* $\mathbf{t}_1, \mathbf{t}_2,$ *and* $\mathbf{t}_3$ *such that*

$$\mathbf{t} = (\mathbf{t}_1 \oplus \mathbf{t}_2) + \mathbf{t}_3,$$

where $\mathbf{t}_1$ *is a 3* $\times$ *3 invariant template and* $\mathbf{t}_2$ *and* $\mathbf{t}_3$ *are* $(2m-1)\times(2m-1)$ *totally symmetric variant templates.*

PROOF:

If the corner values of $\mathbf{t}$ are zero, then each boundary has either $(2m-1)$ or $(2n-1)$ elements which form the boundaries of a $(2m-1)\times(2n-1)$ template $\mathbf{t}_2$ symmetric with respect to both axes.

If $\mathbf{t}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & <0> & 1 \\ 0 & 1 & 0 \end{bmatrix}$, then the template $\mathbf{t}_3 = \mathbf{t} - \mathbf{t}_1 \oplus \mathbf{t}_2$ is a $(2m-1)\times$ $(2n-1)$ template symmetric with respect to both axes.

Q.E.D.

CHAPTER 5

EXAMPLES

In the examples presented below, we have indicated the value in the center pixel location by $< x >$. Except for Example 6, all templates are shift-invariant.

Example 1.

The template

$$\mathbf{h} = \begin{bmatrix} -13 & 2 & 7 & 2 & -13 \\ 2 & 17 & 22 & 17 & 2 \\ 7 & 22 & <27> & 22 & 7 \\ 2 & 17 & 22 & 17 & 2 \\ -13 & 2 & 7 & 2 & -13 \end{bmatrix}$$

(used by Haralick [4]) satifies the conditions of Corollary 2 to Theorem 3.1. Using the techniques of Corollary 2 it has the following decomposition

$$\begin{bmatrix} -13 & -19.73 & -13 \\ -19.73 & <-28.9433> & -19.73 \\ -13 & -19.73 & -13 \end{bmatrix} \oplus \begin{bmatrix} 1 & -1.672 & 1 \\ -1.672 & <1.5411> & -1.672 \\ 1 & -1.672 & 1 \end{bmatrix} + [8.39].$$

Using the rank method (note that $\mathbf{h}$ has rank 2), the template $\mathbf{h}$ can be decomposed as

$$\mathbf{h} = \begin{bmatrix} 1 & 1.5275 & 1 \\ 1.5282 & <2.3191> & 1.5282 \\ 1 & 1.5275 & 1 \end{bmatrix} \oplus \begin{bmatrix} -13 & 19.8575 & -13 \\ 21.7364 & <-33.2024> & 21.7364 \\ -13 & 19.8575 & -13 \end{bmatrix}$$

$$+ \begin{bmatrix} 1 & 1.3333 & 1 \\ 1.2396 & <1.6528> & 1.2396 \\ 1 & 1.3333 & 1 \end{bmatrix} \oplus \begin{bmatrix} 2 \\ <14.5208> \\ 2 \end{bmatrix}.$$

Using the LU factorization, the template $\mathbf{h}$ can be decomposed into

$$\mathbf{h} = \begin{bmatrix} 1 & 1.5182 & 1 \\ 1.5182 & <2.3049> & 1.5182 \\ 1 & 1.5182 & 1 \end{bmatrix} \oplus \begin{bmatrix} -13 & 21.7364 & -13 \\ 21.7364 & <-36.3433> & 21.7364 \\ -13 & 21.7364 & -13 \end{bmatrix}$$

$$+ \begin{bmatrix} 17.3077 & 23.0769 & 17.3077 \\ 23.0769 & <30.7692> & 23.0769 \\ 17.3077 & 23.0769 & 17.3077 \end{bmatrix}.$$

Thus, the first decomposition provides a more efficient implementation of $\mathbf{h}$ than the other two decompositions.

Example 2.

Using Corollary 3.2, the template

$$\mathbf{s} = \begin{bmatrix} -5 & -4 & 0 & 4 & 5 \\ -8 & -10 & 0 & 10 & 8 \\ -10 & -20 & <0> & 20 & 10 \\ -8 & -10 & 0 & 10 & 8 \\ -5 & -4 & 0 & 4 & 5 \end{bmatrix}$$

can be decomposed into the form

$$\mathbf{s} = \begin{bmatrix} -5 & -4 & -5 \\ -8 & <-10> & -8 \\ -5 & -4 & -5 \end{bmatrix} \oplus \begin{bmatrix} 1 & 0 & -1 \\ 0 & <0> & 0 \\ 1 & 0 & -1 \end{bmatrix} + [-12 \quad <0> \quad 12].$$

Note that the template $\mathbf{s}$ is skew symmetric with respect to the $\mathbf{y}$-axis and symmetric with respect to the $\mathbf{x}$-axis.

Example 3.

The Laplace template

$$l = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & <4> & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

satifies the conditions of Proposition 3.6. It can be decomposed into

$$l = \begin{bmatrix} 1 & 0 & 0 \\ 0 & <0> & 0 \\ 0 & 0 & 1 \end{bmatrix} \oplus \begin{bmatrix} 0 & 0 & -1 \\ 0 & <0> & 0 \\ -1 & 0 & 0 \end{bmatrix} + [4].$$

<u>Example 4.</u>

The template

$$\mathbf{t} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & <0> & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

cannot be decomposed into the form $(\mathbf{t}_1 \oplus \mathbf{t}_2) + \mathbf{t}_3$, where each $\mathbf{t}_i$ is a 3×3 operator. Using the LU factorization or the rank method, $\mathbf{t}$ can still be decomposed as the sum and product of four 3×3 operators. Note that this template is symmetric with respect to the $\mathbf{y}$-axis. Thus, the hypothesis in Corollary 1 to Theorem 3.1 that the corner values are different from zero is necessary.

Example 5.

The template

$$t = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & <1> & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

cannot be decomposed into the form $(\mathbf{t}_1 \oplus \mathbf{t}_2) + (\mathbf{t}_3 \oplus \mathbf{t}_4)$, where each $\mathbf{t}_i$ is a shift-invariant 3×3 operator. Note that this template is symmetric with respect to the y-axis.

Example 6.

Let $\mathbf{X} = \{(1,1),(1,2),(2,1),(2,2)\}$, and let $\mathbf{t}$ be the variant template defined by the rules

$$\mathbf{t}(1,1) = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \qquad \mathbf{t}(1,2) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix},$$

$$\mathbf{t}(2,1) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \text{ and } \mathbf{t}(2,2) = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}.$$

The matrix $M_{\mathbf{t}}$ associated with $\mathbf{t}$ is given by

$$M_{\mathbf{t}} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Note that while the template $\mathbf{t}$ has rank one for all $\mathbf{x}$ in $\mathbf{X}$, it fails to be separable.

CHAPTER 6

OPERATOR INVERSION

6.1. Introduction

Inverse problems have come to play a central role in modern applied mathematics, such as in mathematical physics, in imaging areas such as tomography, remote sensing, and restoration. Rosenfeld and Kak [30] explained the equivalence between techniques such as the Wiener filter and the least square methods and the problem of inverting a block circulant matrix with circulant blocks. In 1973, G.E.Trapp [35] showed how the Discrete Fourier Transform could be used to diagonalize and invert a matrix that was either circulant or block circulant with circulant blocks. While the algebraic relationship between circulant matrices and polynomials was completely formulated by J.P. Davis [6], it was P.D.Gader and G.X.Ritter [9,27] who made the connection between polynomials and circulant templates. A circulant template defined on a rectangular array with m rows and n columns is invertible if and only if its corresponding polynomial $p(x,y)$ has the property that $p(\omega_n^j, \omega_m^k) \neq 0$, for all $0 \leq j \leq n$, and $0 \leq k \leq m$. (The symbol ω_n denotes the root of the unity, $\exp(2\pi i/n)$.)

An application of this method is that the usual 3×3 mean filter defined on a rectangular array with m rows and n columns is invertible if and only if the number

3 does not divide either m or n. Gader asked if a similar result applies to the von Neumann mean filter. The von Neumann mean filter is the template $\mathbf{t}$ defined by

$$\mathtt{t} = \begin{array}{|c|c|c|} \cline{2-2} \multicolumn{1}{c|}{} & 1 & \multicolumn{1}{c}{} \\ \hline 1 & <1> & 1 \\ \hline \multicolumn{1}{c|}{} & 1 & \multicolumn{1}{c}{} \\ \cline{2-2} \end{array}\ .$$

The main result of this chapter is to prove that if $\mathbf{t}$ is defined on a square array with m rows and m columns, then $\mathbf{t}$ is invertible if and only if neither 5 nor 6 divide m.

In this section, we will briefly describe the relationship between polynomial algebra and image algebra. More precisely, the relationship between circulant templates and quotient rings of polynomial rings. Important results, such as how to decompose circulant templates and how to obtain minimal local decompositions of separable circulant templates can be found in Gader [8].

6.2. The von Neumann Template

6.2.1. Preliminaries

DEFINITION 6.1. *A $m \times m$ matrix $C = c(i,j)$ is said to be circulant if and only if for every $s \in \mathbf{Z}$, $c_{ij} = c_{(i+s)(mod\ \ m),(j+s)(mod\ \ m)}$.*

Thus, C is a matrix of the form $\begin{bmatrix} c_0 & c_1 & c_2 & \dots & c_{m-1} \\ c_{m-1} & c_0 & c_1 & \dots & c_{m-2} \\ \dots & \dots & \dots & \dots & \dots \\ c_2 & c_3 & c_4 & \dots & c_1 \\ c_1 & c_2 & c_3 & \dots & c_0 \end{bmatrix}$.

Each row is just the previous row cycled forward one step, so that the entries in each row are just a cyclic permutation of those in the first row. We will write $C = \mathrm{circ}(c_0, c_1, ..., c_{m-1})$.

The $m \times m$ permutation matrix $P = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix}$ is called the basic permutation matrix.

A circulant matrix $C = \mathrm{circ}(c_0, c_1, ..., c_{m-1})$, can be associated with the polynomial $p_C(x) = c_0 + c_1 x + ... + c_{m-1} x^{m-1}$. If $C = \mathrm{circ}(c_0, c_1, ..., c_{m-1})$, then $C = p_C(P)$. Because of this representation, circulant matrices have a very nice structure which can be related to P. Since a matrix is circulant if and only if it commutes with the matrix P, the product of two circulants is again a circulant. Thus, the set $\mathbf{C}_m$ of all $m \times m$ circulant matrices is a subring of the set of all matrices of order m. Furthermore, circulant matrices commute under multiplication.

Let $\mathbf{R}[x]$ denote the ring of polynomials in one variable with coefficients in $\mathbf{R}$, and $\mathbf{R}[x]/(x^m - 1)$ the quotient ring of polynomials modulo $x^m - 1$. Multiplication of elements of this ring are performed by replacing x^m by 1.

If we define the mapping $\eta : \mathbf{C}_m \rightarrow \mathbf{R}[x]/(x^m - 1)$, by $\eta(C) = p_C(x)$, then η is a ring isomorphism [6].

DEFINITION 6.2. *Let B be an $mn \times mn$ matrix. We say that B is block circulant with circulant blocks of type (m, n) if and only if there exist $n \times n$ circulant matrices $C_0, C_1, ..., C_{m-1}$ such that $B = circ(C_0, C_1, ..., C_{m-1})$*

As in the circulant case, given a block circulant matrix of the form $C = \mathrm{circ}(c_0, c_1, ..., c_{mn-1})$, with circulant blocks of type (m, n), we can associate a polynomial $p_C(x, y)$ in two variables as follows:

$$p_C(x,y) = (c_0 + c_1 y + ... + c_{n-1} y^{n-1}) + x(c_n + c_{n+1} y + ... + c_{2n-1} y^{n-1}) + \\ ... + x^{m-1}(c_{(m-1)n} + c_{(m-1)n+1} y + ... + c_{mn-1} y^{n-1}).$$

As in the one variable case, $\mathbf{R}[x, y]/(x^m - 1, y^n - 1)$ denotes the quotient ring of polynomials modulo $x^m - 1$, and $y^n - 1$.

If we define the mapping $\theta : \mathbf{C}_{mn} \rightarrow \mathbf{R}[x, y]/(x^m - 1, y^n - 1)$ by $\theta(C) = p_C(x, y)$, then θ is a ring isomorphism [6].

Therefore, given a circulant matrix $\mathbf{M} \in \mathbf{C}_{mn}$, $\mathbf{M}$ is invertible if and only if $\theta(\mathbf{M}) = p_{\mathbf{M}}(x, y)$ is a unit in $\mathbf{R}[x, y]/(x^m - 1, y^n - 1)$.

Let $\omega_m^j = \exp(-2\pi i j/m)$, where $i = \sqrt{-1}$, and let Λ be the diagonal matrix $\mathrm{diag}(\omega_m^j)$, for $j = 0, 1, ..., m - 1$. It can be proved that the permutation matrix described above, satisfies the equation $P = F \Lambda F^*$, where F is the one dimensional Discrete Fourier Transform of order m, and * denotes the conjugate transpose. Since F is a unitary matrix, we have that $F^* = F^{-1}$. Therefore, if C is circulant, then

$$C = p_C(P) = p_C(F \Lambda F^*) = F p_C(\Lambda) F^* = F \Delta F^*,$$

where Δ is the diagonal matrix, $\mathrm{diag}(p_C(\omega_m^j))$, for $j = 0, 1, ..., m-1$, whose diagonal

terms are the Fourier coefficients. Thus, any circulant matrix can be diagonalized by the Fourier matrix.

Similarly, if F is the $mn \times mn$ two-dimensional Discrete Fourier matrix, and $\mathbf{M}$ is a block circulant matrix with circulant blocks, then we can write $\mathbf{M} = FDF^*$, where F is the $mn \times mn$ two-dimensional Fourier matrix, and D is a diagonal matrix.

Therefore, if the Fourier coefficients of $\mathbf{M}$ are all nonzero, then D is invertible and the inverse of $\mathbf{M}$ is given by $\mathbf{M}^{-1} = FD^{-1}F^*$ [6].

Let $\mathbf{t}$ be an invariant circulant template. The following theorem has been proved in Gader [8] .

THEOREM 6.3. *Let* $\mathbf{t}$ *be a circulant template on the coordinate set* $\mathbf{X}$. *The corresponding matrix* $\mathbf{M_t}$ *is a block circulant matrix with circulant blocks.*

Since we can write $\mathbf{M_t} = FDF^*$, where $D = \text{diag}(p_{\mathbf{t}}(\omega_m^j, \omega_n^k))$ for $j = 0, ..., m-1$, and $k = 0, ..., n-1$, $\mathbf{M_t}$ is invertible if and only if $p_{\mathbf{t}}(\omega_m^j, \omega_n^k) \neq 0$, for every j and k.

Example: Let $\mathbf{t}$ be the template defined on the $m \times n$ array $\mathbf{X}$ by

$$\mathbf{t}(i,j) = \{(i,j),\ (i+1(mod\ \ m),j),\ (i-1(mod\ \ m),j),\ (i,j+1(mod\ \ n))\ (i,j-1(mod\ \ n))\}.$$

$\mathbf{t}$ is called the *von Neumann* averaging (or mean) template. This template can also be defined by the rule

$$\mathbf{t}_{(i,j)}(r,s) = \begin{cases} 1 & \text{if } |i-k| + |j-l| \leq 1 \\ 0 & \text{otherwise} \end{cases}.$$

The polynomial associated with this template, via the isomorphism θ, is

$$p_{\mathbf{t}}(x,y) = 1 + x + x^{n-1} + y + y^{m-1}.$$

Let $\mathbf{X}$ be a rectangular array with m rows and n columns, and let $\mathbf{E}$ be the template defined by :

$$\mathrm{E}(\mathrm{x}) = \{(\mathrm{y}, \mathrm{e}_{\mathrm{x}}(\mathrm{y})) : \mathrm{e}_{\mathrm{x}}(\mathrm{x}) = 1 \ \text{ and } \quad \mathrm{e}_{\mathrm{x}}(\mathrm{y}) = 0 \quad \text{if } \ \mathrm{y} \neq \mathrm{x}\}.$$

$\mathbf{E}$ is the identity template for the convolution $\oplus$; that is

$$\mathbf{t} \oplus \mathbf{E} = \mathbf{E} \oplus \mathbf{t} = \mathbf{t}.$$

A template $\mathbf{t}$ on $\mathbf{X}$ is said to be *invertible* with respect to $\oplus$ if there exists a template $\mathbf{s}$ on $\mathbf{X}$ such that

$$\mathbf{t} \oplus \mathbf{s} = \mathbf{s} \oplus \mathbf{t} = \mathbf{E}.$$

PROPOSITION 6.4. *The von Neumann averaging template* $\mathbf{t}$, *defined on a finite rectangular coordinate set with* m *rows and* n *columns, is invertible if and only if* $p_{\mathbf{t}}(\omega_m^j, \omega_n^k) \neq 0$, *for every* j *and* k.

6.2.2. The Main Theorem

THEOREM 6.5. *If* $\mathbf{X}$ *is an* $m \times m$ *rectangular coordinate set in* $\mathbf{Z} \times \mathbf{Z}$, *and if* $\mathbf{t}$ *denotes the von Neumann mean filter template defined on* $\mathbf{X}$, *then* $\mathbf{t}$ *is invertible if and only if neither 5 nor 6 divide* m.

PROOF:

For $j, k = 0, ..., m-1$ we have

$$p_{\mathbf{t}}(\omega_m^j, \omega_m^k) = 1 + \omega_m^j + \omega_m^{-j} + \omega_m^k + \omega_m^{-k}$$

$$= 1 + 2cos(2\pi j/m) + 2cos(2\pi k/m).$$

For convenience, we will write ω instead of ω_m.

We first assume that one of the integers 5 or 6 divides m.

Let $m = 5d$, for some d. If we let $j = d$ and $k = 2d$, then

$$p_{\mathbf{t}}(\omega^j, \omega^k) = 1 + \omega^d + \omega^{-d} + \omega^{2d} + \omega^{-2d}$$

$$= 1 + \omega_5 + \omega_5^{-1} + \omega_5^2 + \omega_5^{-2}$$

$$= 1 + \omega_5 + \omega_5^2 + \omega_5^3 + \omega_5^4 = 0$$

for this is the sum of the five fifth roots of unity.

Let $m = 6d$, for some d. If we let $j = d$ and $k = 3d$, then

$$p_{\mathbf{t}}(\omega^j, \omega^k) = 1 + \omega^d + \omega^{-d} + \omega^{3d} + \omega^{-3d}$$

$$= 1 + 2cos(\pi) + 2cos(\pi/3) = 0.$$

Hence if 5 or 6 divides m, there exist two integers j and k such that $p_{\mathbf{t}}(\omega_m^j, \omega_m^k)$ $= 0$ and, by Proposition 2.4, the template $\mathbf{t}$ is not invertible.

WE now prove the sufficient condition.

Due to the symmetry of j and k, in the polynomial expression $p_{\mathbf{t}}(\omega^j, \omega^k)$, it is enough to restrict these integers to $0 \leq j \leq k \leq m/2$.

Let j be the minimal value such that $p_{\mathbf{t}}(\omega^j, \omega^k) = 0$.

Note that if

$$q(x) = 1 + x^j + x^{(m-1)j} + x^k + x^{(m-1)k},$$

then the equation $p_{\mathbf{t}}(\omega^j, \omega^k) = 0$, is equivalent to $q(\omega) = 0$. If we use the fact that cyclotomic polynomials are the minimal polynomials of the primitive roots of unity [3], then we have that $q(\omega^s) = 0$, for any s relatively prime to m.

Claim: If $p_t(\omega^j, \omega^k) = 0$, then $m/6 \leq j \leq k \leq m/2$.

If for any $z = r{*}\exp(i\alpha)$, $Arg(z) = \alpha$, then $Arg(\omega^j) = Arg(\omega^{-j})$. Note that in order to find two integers j and k such that $p_t(\omega_m^j, \omega_m^k) = 0$, it is necessary that $\pi/3 \leq Arg(\omega^j) \leq \pi$. Since $Arg(\omega^j) = 2\pi j/m$, the previous argument gives $m/6 \leq j \leq m/2$. The claim is proved.

We will consider two cases: j divides m and j does not divide m.

Case 1: j divides m.

Using the claim, we know that $2j \leq m \leq 6j$. If j divides m, then m is of the form $2j, 3j, 4j, 5j$ or $6j$.

If $m = 2j$, then $p_t(\omega^j, \omega^k) = 0$implies that $cos(2\pi k/m) = -1/2$, which implies that $k = m/3$. Since k is an integer, 3 divides m. Hence 6 divides m.

If $m = 3j$, then $p_t(\omega^j, \omega^k) = 0$ implies that $cos(2\pi k/m) = 0$, which implies that $k = m/6$. Since 3 and 4 divide m, 6 divides m.

If $m = 4j$, then $p_t(\omega^j, \omega^k) = 0$ implies that $cos(2\pi k/m) = -1/2$, which implies that $k = m/3$. Since 3 and 4 divide m, 6 divides m.

If $m = 5j$ or $6j$, then since j is an integer, we conclude that 5 or 6 divides m.

Case 2: j does not divide m.

Using the claim again, we can write m as $2j + r, 3j + r, 4j + r$, or $5j + r$ for some nonzero $r < j$.

If $m = 3j + r$, then either 2 divides m or 2 does not divide m. In the case where 2 divides m, if 3 divides m, we are done. Otherwise, the greatest commun divisor of m and 3, denoted by gcd(m,3), is equal to 1. Thus

$$q(\omega^3) = 1 + \omega^{3j} + \omega^{-3j} + \omega^{3k} + \omega^{-3k} = 0.$$

But $3j = m - r$, thus,

$$q(\omega^3) = 1 + \omega^{m-r} + \omega^{-(m-r)} + \omega^{3k} + \omega^{-3k}$$
$$= 1 + \omega^{r} + \omega^{-r} + \omega^{3k} + \omega^{-3k},$$

which contradicts the minimality of j. In the case where 2 does not divide m, then gcd(m,4)=1, and

$$q(\omega^4) = 1 + \omega^{4j} + \omega^{-4j} + \omega^{4k} + \omega^{-4k} = 0.$$

But, $4j = m - r + j$, thus,

$$q(\omega^4) = 1 + \omega^{m-r+j} + \omega^{-(m-r+j)} + \omega^{4k} + \omega^{-4k}$$
$$= 1 + \omega^{j-r} + \omega^{-(j-r)} + \omega^{4k} + \omega^{-4k},$$

which contradicts the minimality of j.

If $m = 4j + r$, then either 3 divides m or 3 does not divide m. In the case where 3 divides m, if 2 divides m we are done. Otherwise, gcd(m,4)=1. Thus,

$$q(\omega^4) = 1 + \omega^{4j} + \omega^{-4j} + \omega^{4k} + \omega^{-4k} = 0.$$

But, $4j = m - r$, thus,

$$q(\omega^4) = 1 + \omega^{m-r} + \omega^{-(m-r)} + \omega^{4k} + \omega^{-4k}$$
$$= 1 + \omega^{r} + \omega^{-r} + \omega^{5k} + \omega^{-5k},$$

which contradicts the minimality of j. In the case where 3 does not divide m, then gcd(m,9)=1 and

$$q(\omega^9) = 1 + \omega^{9j} + \omega^{-9j} + \omega^{9k} + \omega^{-9k} = 0.$$

But $9j = 2m - 2r + j$, thus,

$$q(\omega^9) = 1 + \omega^{2m-2r+j} + \omega^{-(2m-2r+j)} + \omega^{9k} + \omega^{-9k}$$
$$= 1 + \omega^{j-2r} + \omega^{-(j-2r)} + \omega^{9k} + \omega^{-9k},$$

which contradicts the minimality of j, since $|j - 2r| \leq j$.

If $m = 5j + r$, then if 5 divides m we are done. Otherwise, gcd(m,5) = 1 and

$$q(\omega^5) = 1 + \omega^{5j} + \omega^{-5j} + \omega^{5k} + \omega^{-5k} = 0,$$

since ω^5 is a primitive m^{th} root of unity. But $5j = m - r$, thus,

$$
\begin{aligned}
q(\omega^5) &= 1 + \omega^{m-r} + \omega^{-(m-r)} + \omega^{5k} + \omega^{-5k} \\
&= 1 + \omega^{r} + \omega^{-r} + \omega^{5k} + \omega^{-5k},
\end{aligned}
$$

which contradicts the minimality of j, since $r \leq j$. This concludes the proof of Theorem 3.1

Q.E.D.

A generalization of the von Neumann averaging template is the one defined by

$$
\mathtt{t}_{(i,j)}(r,s) = \begin{cases} a & \text{if } r = i, \text{ and } s = j \\ b & \text{otherwise} \end{cases},
$$

$$
\mathtt{t} = \begin{array}{|c|c|c|}
\cline{2-2}
\multicolumn{1}{c|}{} & \mathtt{b} & \multicolumn{1}{c}{} \\
\hline
\mathtt{b} & \mathtt{<a>} & \mathtt{b} \\
\hline
\multicolumn{1}{c|}{} & \mathtt{b} & \multicolumn{1}{c}{} \\
\cline{2-2}
\end{array},
$$

where a, and b are two real numbers.

The associated polynomial is $p(x,y) = a + bx + bx^{m-1} + by + by^{m-1}$, and

$$
\begin{aligned}
p(\omega^j, \omega^k) &= a + b\omega^j + b\omega^{-j} + b\omega^k + b\omega^{-k} \\
&= a + b(\omega^j + \omega^{-j} + \omega^k + \omega^{-k}).
\end{aligned}
$$

PROPOSITION 6.6. *If $a = 0$, then the template* $\mathtt{t}$ *is invertible if and only if* m *is an odd number.*

PROOF:

Of course, we will assume, here, that $b \neq 0$, in which case it can be factored.

If
$$
\begin{aligned}
p(\omega^j, \omega^k) &= \omega^j + \omega^{-j} + \omega^k + \omega^{-k} \\
&= 2cos(2\pi j/m) + 2cos(2\pi k/m) = 0,
\end{aligned}
$$

then the equation is equivalent to $cos(2\pi j/m) = -cos(2\pi k/m)$, which implies $2\pi j/m + 2\pi k/m = \pi$, and therefore, $j + k = m/2$. Since j and k are integers, this proves that for any j, and $k = m/2 - j$, $p(\omega^j, \omega^k) = 0$. Therefore, the template **t** is not invertible if m is an even number.

Q.E.D.

PROPOSITION 6.7. *If $a \neq 0$ and $|a/b| > 4$, then the template* **t** *is invertible.*

PROOF:

If
$$p(\omega^j, \omega^k) = a + b\omega^j + b\omega^{-j} + b\omega^k + b\omega^{k-1}$$
$$= a + 2b(cos(2\pi j/m) + 2cos(2\pi k/m)) = 0,$$

then $|cos(2\pi j/m) + cos(2\pi k/m)| = |a/2b| > 2$, which is a contradiction. Therefore, $p(\omega^j, \omega^k)$ is always nonzero, and the template **t** is invertible.

Q.E.D.

6.3. The Characteristic Polynomial Method

Let **X** be a coordinate set with m rows and n columns, and let **t** be a template defined on **X**. Using the relationship between the matrix algebra and the image algebra, under the isomorphism ψ, we know that the corresponding matrix $\psi(\mathbf{t}) = M_{\mathbf{t}}$ is an $mn \times mn$ block matrix. There are m^2 blocks and each block is an $n \times n$ matrix.

The determinant $\det(M_{\mathbf{t}} - \lambda I_{mn})$, where I_{mn} is the unit $mn \times mn$ unit or identity matrix, is called the characteristic polynomial of $M_{\mathbf{t}}$. It has the form

$$p(\lambda) = \lambda^{mn} + a_{mn-1}\lambda^{mn-1} + ... + a_1\lambda + a_0.$$

In $p(\lambda)$, the constant a_0 is the determinant of $M_{\mathbf{t}}$, while the coefficient a_{mn-1} is the trace of $M_{\mathbf{t}}$ [12].

THEOREM 6.8. *If a_0 is nonzero, then $M_{\mathbf{t}}$ is invertible and its inverse is given by the formula:*

$$M_{\mathbf{t}}^{-1} = \frac{-1}{a_0}(M_{\mathbf{t}}^{mn-1} + a_{mn-1}M_{\mathbf{t}}^{mn-2} + ... + a_1 I_{mn}).$$

PROOF:

By the Cayley-Hamilton theorem, we know that the matrix $M_{\mathbf{t}}$ satisfies its characteristic polynomial $p(\lambda)$, i.e $p(M_{\mathbf{t}}) = 0$. Therefore,

$$M_{\mathbf{t}}^{mn} + a_{mn-1}M_{\mathbf{t}}^{mn-1} + ... + a_1 M_{\mathbf{t}} + a_0 I_{mn} = 0.$$

The previous equation can be rewritten as

$$M_{\mathbf{t}}^{mn} + a_{mn-1}M_{\mathbf{t}}^{mn-1} + ... + a_1 M_{\mathbf{t}} = -a_0 I_{mn}, \text{ which implies}$$

$$M_{\mathbf{t}}[\tfrac{-1}{a_0}(M_{\mathbf{t}}^{mn-1} + a_{mn-1}M_{\mathbf{t}}^{mn-2} + ... + a_1 I_{mn})] = I_{mn}.$$

This last equation gives

$$M_{\mathbf{t}}^{-1} = \tfrac{-1}{a_0}(M_{\mathbf{t}} mn - 1 + a_{mn-1}M_{\mathbf{t}}^{mn-2} + ... + a_1 I_{mn}).$$

Q.E.D.

Let **E** be the identity template defined in Section 6.2.1, and let $\mathbf{t}^n$ be defined as $\mathbf{t} \oplus \mathbf{t} \oplus ... \oplus \mathbf{t}$ n times.

From the previous theorem, we can conclude that, given a shift invariant template $\mathbf{t}$, we can write its inverse as

$$\mathbf{t}_{-1} = \frac{-1}{a_0}(\mathbf{t}^{mn-1} + a_{mn-2}\mathbf{t}^{mn-2} + ... + a_1\mathbf{E}).$$

Since the inverse of a block toeplitz matrix with toeplitz blocks is not necessarily a matrix with the same properties, we see that even the inverse of a very small size shift invariant operator can be (and most probably will be) a full size variant operator. The previous theorem is one way to write the inverse of a shift invariant template in terms of the template itself and therefore, in terms of invariant templates.

6.4. Inverse of General Shift Invariant Operators

In this section, we discuss some properties of block Toeplitz matrices with Toeplitz blocks as they are directly related to shift invariant templates which are, as mentioned previously, used to implement convolutions. The idea here is to apply some results on Toeplitz matrices to the image algebra.

As in the beginning of this chapter, the matrices of this section are indexed by sets of the form $\{0, 1, ..., m-1\}$.

Definition 6.9. *Let $M = (m_{ij})$ be an $m \times m$ matrix. We say that M is a Toeplitz matrix if and only if for every $i, j \in \{0, 1, ..., m-1\}$, and k in $\mathbf{Z}$ such that $i+k, j+k \in \{0, 1, ..., m-1\}$, we have $a_{ij} = a_{i+k,j+k}$.*

Thus, M is a matrix of the form $\begin{bmatrix} a_0 & a_1 & \cdots & a_{m-1} \\ a_{-1} & a_0 & \cdots & a_{m-2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{-(m-1)} & a_{-(m-2)} & \cdots & a_0 \end{bmatrix}$.

A Toeplitz matrix is constant along any diagonal and has its ij^{th} entry as a function of $(i-j)$ rather than of i and j separately.

DEFINITION 6.10. *Let $M = (M_{ij})$ be an $m \times m$ block matrix, where each M_{ij} is an $n \times n$ matrix. We say that M is block Toeplitz with Toeplitz blocks if and only if M is of the form*

$$\begin{bmatrix} A_0 & A_1 & \cdots & A_{m-1} \\ A_{-1} & A_0 & \cdots & A_{m-2} \\ \cdots & \cdots & \cdots & \cdots \\ A_{-(m-1)} & A_{-(m-2)} & \cdots & A_0 \end{bmatrix},$$

where for every i in $\{-(m-1),(m-1)\}$, each A_i is a Toeplitz matrix.

The following theorem was proved in Gader [8].

THEOREM 6.11. *Let $\mathbf{X}$ be a rectangular coordinate set. If $\mathbf{t}$ is a template on $\mathbf{X}$, and $\mathbf{M_t}$ is the corresponding matrix under ψ, then $\mathbf{M_t}$ is a block Toeplitz matrix with Toeplitz blocks.*

Since the inverse of a Toeplitz matrix is not necessarily a matrix of the same form, we see that this very special structure of Toeplitz and block Toeplitz matrices strongly suggests that an inversion scheme exploiting this structure would yield significant savings in time and effort. Several researchers have tried to investigate Toeplitz matrices and their inverses. Some algorithms like the Trench algorithm proved to be powerful [33]. Other methods yielded very elegant formulas for the

inverse [1].

Let us first give some facts pertinent only to Toeplitz matrices.

Let M be an $m \times m$ Toeplitz matrix; u, v, two $m \times 1$ vectors; and e^i the transpose of the vector $(0, ...0, 1, 0, ...0)$, where 1 is at the i^{th} location.

Fact 1: If $M(u_0, u_1, ..., u_{m-1})^t = (v_0, v_1, ..., v_{m-1})^t$,

then $(u_{m-1}, u_{m-2}, ..., u_0)M = (v_{m-1}, v_{m-2}, ..., v_0)$.

Fact 2: If $Mu = e^p$, and $Mv = e^q$,

then $u_{m-1-q} = v_{m-1-p}$.

Note that we have

$$[(v_{m-1}, ..., v_0)M](u_0, ..., u_{m-1})^t = (v_{m-1}, ..., v_0)[M(u_0, ..., u_{m-1})].$$

We will call a block vector of dimension m, a vector in which all entries are $n \times n$ matrices. It will be denoted by a bold letter. We will denote by $\mathbf{e_i}$ the vector $(0, ..., 0, I, 0, ..., 0)$, where the $n \times n$ unit or identity matrix I is at the i^{th} entry, and where 0 stands for the $n \times n$ zero matrix. $L(A_0, A_1, ..., A_{m-1})$ represents the lower triangular block Toeplitz matrix

$$L = \begin{bmatrix} A_0 & 0 & \dots & 0 \\ A_1 & A_0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ A_{(m-1)} & A_{-(m-2)} & \dots & A_0 \end{bmatrix}.$$

In a manner similar to the lower triangular matrix, we denote the upper triangular block Toeplitz matrix by $U(A_0, A_1, ..., A_{m-1})$ and finally, we let the matrix $\mathbf{S}$ denote the *shift* matrix $\mathbf{S} = L(0, I, 0, ..., 0)$.

PROPOSITION 6.12. *Let* $\mathbf{M}$ *be an* $m \times m$ *block matrix with* $n \times n$ *blocks. If there exist column block vectors* $\mathbf{x}$, *and* $\mathbf{y}$ *such that*

$$\mathbf{Mx} = \mathbf{e}_0 \quad \textit{and} \quad \mathbf{My} = \mathbf{SMe}_m,$$

then $\mathbf{M}$ *is invertible. Moreover, let* $\mathbf{w}$ *and* $\mathbf{z}$ *be two block vectors such that*

$$\mathbf{wM} = \mathbf{e}_n^t \quad \textit{and} \quad \mathbf{zMy} = \mathbf{e}_0^t\mathbf{MS},$$

then

$$\mathbf{M}^{-1} = \mathbf{A}_1\mathbf{B}_1 - \mathbf{A}_2\mathbf{B}_2,$$

where

$$\mathbf{A}_1 = L(\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_{m-1}) \ \ , \mathbf{A}_2 = L(-\mathbf{y}_0, -\mathbf{y}_1, ..., -\mathbf{y}_{m-1}),$$

$$\mathbf{B}_1 = U(\mathbf{I}, -\mathbf{z}_0, ..., -\mathbf{z}_{m-2}) \ \ , \mathbf{B}_2 = U(0, \mathbf{w}_0, ..., \mathbf{w}_{m-2}).$$

In the implementation of this theorem, we transpose the dreadful problem of inverting an $mn \times mn$ block Toeplitz matrix with Toeplitz blocks to the problem of solving $4n$ linear systems of mn equations with mn variables.

In the application of this theorem to image algebra, the result is quite interesting. If $\mathbf{t}$ is an invariant template satisfying the conditions of the theorem, then there exist four templates $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$, and $\mathbf{t}_4$ such that

$$\mathbf{t}^{-1} = \mathbf{t}_1 \oplus \mathbf{t}_2 + \mathbf{t}_3 \oplus \mathbf{t}_4,$$

where $\mathbf{t}_1$, and $\mathbf{t}_3$ are shift invariant along each row, and $\mathbf{t}_2$ and $\mathbf{t}_4$ are shift invariant along each column

CHAPTER 7

CONCLUSION AND SUGGESTIONS FOR FURTHER RESEARCH

This dissertation has addressed the problems of template decomposition and template inversion. The framework is based on the image algebra and its relationship to polynomial and matrix algebra.

In particular, it was shown that:

1) A two-dimensional shift invariant operators defined on a rectangular array can be decomposed into sums and products of smaller size operators. The decomposition methods presented here, are suitable for a machine with a pipeline architecture.

2) Two-dimensional shift invariant operators exhibiting some kind of symmetry require a much smaller number of operators in the decomposition. We focused our attention on this type of operator because of their frequent use in image processing.

3) The techniques were extended to those operators which may not be shift invariant. Necessary and sufficient conditions were proved to test for the separability of a variant operator. Necessary and sufficient conditions were proved to test whether or not a variant operator can be written as the sum of separable operators.

4) Necessary and sufficient conditions were given to characterize when the mean filter based on the von Neumann configuration is invertible. Since the inverse

of a shift invariant operator is not necessarily shift invariant (actually, there is very little chance it is shift invariant), methods were provided for writing the inverse in terms of shift invariant operators.

We now list some suggestions for further research.

First, generalize the results obtained in Chapter 3 for two-dimensional shift invariant operators to arbitrary shift invariant operators of different configuration. Circular and convex configurations would be of particular interest. Can the polynomial decomposition method be generalized to these more general operators ?

The decomposition methods for variant operators would certainely be implemented if the cost in computation time was not prohibitive. Is there a better and more direct approach to their decomposiiton ? Many methods given in Chapter 4 involve shift invariant and variant operators in the decomposition of variant operators. A method involving only shift invariant operators will surely be a more efficient one. An important example of a variant template is the Gabor Transform. Can the results presented here be used to decompose this operator.

A generalization of the condition of the invertibility of the von Neumann mean filter defined on a rectangular array, rather than on a square array should be proved in the the very near future. The inverse of a shift invariant operator can be a full size variant operator. Methods to write this inverse in terms of an acceptable number of shift invariant operators can find their way in areas such as tomography, remote sensing, and cryptography among others.

REFERENCES

1. A. Ben-Artzi and T. Shalom, *"On the Inverse of Toeplitz and Close to Toeplitz Matrices"*, Linear Algebra and Applications **75** (1986), pp. 171-192.
2. G. Birkhoff and J. Lipson, *"Heterogeneous Algebras"*, J. Combinatorial Theory **8** (1970), pp. 115-133.
3. J. P. Blahut, "Fast Algorithms for Digital Signal Processing," John Wiley and Sons, New York, 1985.
4. P. Buchberger, *"Grobner Bases: An Algorithmic Method in Polynomial Theory"*, Recent Trends in Multidimensional Systems Theory (1985).
5. J. L. Davidson, *"Lattice Structure in the Image Algebra and their Applications" to image processing"*, Ph.D. Dissertation, University of Florida, Gainesville, Fl (1989).
6. J. P. Davis, "Circulant Matrices," John Wiley and Sons, New York, 1979.
7. D. Gabor, *"Theory of Communications"*, J. Inst. Elec. Eng. (London) **93** (1946), pp. 149-157.
8. P. D. Gader, *"Image Algebra Techniques for Parallel Computation of Discrete Fourier Transforms and General Linear Transforms"*, Ph.D. Dissertation, University of Florida, Gainesville, Fl (1986).
9. W. Grobner, "Modern Algebraic Geometry," Springler Verlag, 1949.
10. R. M. Haralick, *"R1idges and Valleys on Digital Images"*, Journal of Computer Vision, Graphics and Image Processing **22** (1983), pp. 28-38.
11. C. E. Heil and D. F. Walnut, *"Continuous and Discrete Wavelets Transforms"*, SIAM Review **31 No 4** (1989), pp. 628-666.
12. N. Jacobson, "Basic Algebra I," Freeman and Company, San Francisco, 1974.
13. D. E. Knuth, "The Art of Computer Programming, Sorting and Searching," Addison Wesley, Reading, MA, 1973.
14. D. E. Knuth and P. B. Bendix, *"Simple Word Problems in Universal Algebra"*, Proc. Computational Problems in Abstract Algebra, Oxford, England (1970), pp.263-298.

15. B. Kutzler and S. Stifter (Grobner), *"Automated Geometry Theorems Proving Using Buchberger's Algorithm"*, Proceedings SYMSAC '86, Wateloo,Canada (1986, ACM publications), pp.209-214.

16. D. Li, *"Recursive Operations in Image Algebra and their Applicaitons to Image Processing"*, Ph.D. Dissertation, University of Florida, Gainesville, Fl (1990).

17. Z. M. Manseur and D. C. Wilson, *"Decomposition Methods for Convolution Operators"*, submitted.

18. G. Matheron, *"Development of a Mathematical Structure for Image Processing"*, Technical report, Optical Division Tech. Report, Perkin-Elmer (1983).

19. P. E. Miller, "Random Sets and Integral Geometry," Wiley, New York, 1975.

20. D. R. Musser, *"Multivariate Polynomial Factorization"*, Journal of the ACM **22(2)** (April, 1975), pp. 291-308.

21. C. L. Nikias, A. P. Chrysafis, and A. N. Venetsanopoulos, *"The LU Decomposition Theorem and its Implications to the Realization of Two-Dimensional Digital Filters"*, IEEE Transactions on Acoustics, Speech and Signal Processing **3** (1985), pp. 694-711.

22. D. P. O'Leary, *"Some Algorithms for Approximating Convolutions"*, Journal of Computer Vision, Graphics and Image Processing **41** (1988), pp. 333-345.

23. G. X. Ritter, *"An Algebra of Images"*, Proc. Conf on Intelligent Systems and Machines (1984), pp. 333-339.

24. G. X. Ritter, *"The Language of Massively Parallel Processing Computers"*, Proc. 1984 S. E. Reg. ACM Conference, Friendly Systems 1984-2000? (1984), pp. 224-231.

25. G. X. Ritter and P. D. Gader, *"Image Algebra Implementation on Cellular Array Computers"*, IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management (1985), pp. 430-438.

26. G. X. Ritter, P. D. Gader, and J. L. Davidson, *"Automated Bridge Detection in FLIR Images "*, In Proceedings of the Eighth International Conference on Pattern Recognition. Paris, France (to appear).

27. G. X. Ritter and P. D. Gader, *"Image Algebra Techniques for Parallel Image Processing"*, Journal of Parallel Distrib. Comput. **4, No 5** (1987), pp. 7-44.

28. G. X. Ritter, M. A. Shrader-Frechette, and J.N. Wilson, *"Image Algebra: A Rigorous and Translucent Way of Expressing all Image Processing Operations"*, In Proc. of the 1987 SPIE Tech. Symp. Southeast on Optics, Elec.-Opt., and Sensors. Orlando, FL, (May 1987), pp. 7-44.

29. G. X. Ritter, J. N. Wilson, and J. L. Davidson, *"Image algebra: An overview"*, Journal of Computer Vision, Graphics and Image processing **49** (1990), pp. 297-331.

30. A. Rosenfeld and A. C. Kak, "Digital Picture Processing," Academic Press, New York, London, 1982.

31. P. C. Sabatier, "Inverse Problems: An Interdisciplinary Study," Academic Press, New York, 1987, pp. pp. 21-67, 99-175, 217-224.

32. J. Serra,, "Image Analysis," Academic Press, London, 1982.

33. S. R. Sternberg, *"Language and Architecture for Parallel Image Processing"*, Proc. of the conf. on Pattern Recognition in Practice, Amsterdam (1980).

34. S. R. Sternberg, *"Biomedical Image Processing"*, Computer **16(1)** (1983), pp. 22-34.

35. S. R. Sternberg, *"Overview of Image Algebra and Related Issues"*, Integrated Technology for Parallel Image Processing, Academic Press, London (1985).

36. G. E. Trapp, *"Inverses of Circulant Matrices and Block Circulant Matrices"*, Kyungpook Math. J. **13 No 1** (1973), pp. 11-20.

37. W. F. Trench, *"An Algorithm for the Inversion of Finite Toeplitz Matrices"*, Jour. SIAM **12,3** (1964), pp. 515-522.

38. J. N. Wilson, D. C. Wilson and G. X. Ritter, *"The Image Algebra Fortran Language"*, UF-CIS Technical Report TR-88-03,Center for Computer Vision Research, University of Florida (Florida, June 1988).

BIOGRAPHICAL SKETCH

Zohra Z. Manseur was born in Ouenza, eastern Algeria, shortly after the beginning of the Algerian war against France and grew up in Algiers. She received a Licence-es-Sciences of Teaching in Mathematics from the University of Algiers in 1981, and a Diplome d'Etudes Supérieures in Algebra in 1983 from the University of Science and Technology of Algiers. She joined the graduate program in Mathematics at the University of Florida in August 1984 and received a Master's degree in August 1986. She married Rachid Manseur in 1978 and has a son and a daughter.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Gerhard Ritter

Gerhard X. Ritter, Chairman
Professor of Mathematics
Professor of Computer and Information Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate,in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

David C. Wilson

David C. Wilson, CoChairman
Professor of Mathematics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

James E. Keesling

James E. Keesling
Professor of Mathematics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Li-Chien Shen
Associate Professor of Mathematics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Joseph N. Wilson
Assistant Professor of Computer and Information Sciences

This dissertation was submitted to the Graduate Faculty of the Department of Mathematics in the College of Liberal Arts and Sciences and to Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December 1990

Dean, Graduate School